

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

CARRERA: INGENIERÍA DE SISTEMAS

Tesis previa a la obtención del título de: INGENIERO DE SISTEMAS

**TEMA:
DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA PARA EL
REGISTRO Y CONTROL DE TUTORÍAS ACADÉMICAS, BASADO EN UNA
ARQUITECTURA DE TRES CAPAS, PARA LA CARRERA DE INGENIERÍA
EN SISTEMAS DE LA UNIVERSIDAD POLITÉCNICA SALESIANA SEDE
QUITO.**

**AUTOR
DIEGO MANUEL ZURITA GRANIZO**

**DIRECTOR:
WASHINGTON RAÚL PADILLA ARIAS**

Quito, mayo de 2015

DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACION DE USO DEL TRABAJO DE TITULACIÓN

Yo, autorizo a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además, declaro que los conceptos, análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad del autor.

Quito, mayo de 2015

.....

Diego Manuel Zurita Granizo

C.I. 1720296274

DEDICATORIA

A mis padres por ser el eje fundamental de todo lo que soy, en toda mi formación académica y espiritual, por su incondicional apoyo y total sacrificio. Todo esto ha sido posible gracias a ellos.

A Dios por guiar mi camino y darme la sabiduría necesaria para lograr terminar este proceso de mi vida.

A mis hermanos que siempre han sido mi guía durante todos estos años para nunca desfallecer, con mucho cariño familia esto logro lo hemos obtenido juntos.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO I	3
ESTUDIO DE LA SITUACIÓN ACTUAL DE LAS TUTORIAS ACADÉMICAS EN LA CARRERA DE INGENIERÍA DE SISTEMAS.....	3
1.1 Introducción	3
1.2 Problema	4
1.3 Justificación.....	5
1.4 Objetivos	6
1.4.1 Objetivo General.....	6
1.4.2 Objetivos Específicos	6
1.5 Alcance	6
1.6. Marco teórico	8
1.6.1. Tutorías Académicas.....	8
1.6.2 Metodología XP (Extreme Programming).....	9
1.6.3 Arquitectura de tres capas	13
1.7 Situación Actual	13
CAPÍTULO 2	17
ANÁLISIS Y DISEÑO DEL SISTEMA	17
2.1 Especificación del sistema de software	17
2.1.1 Introducción	17
2.2 Descripción general	18
2.2.1 Perspectiva del producto	19
2.2.2 Arquitectura de tres capas	20
2.2.3 Función del producto	22
2.2.4 Limitaciones generales.....	23
2.2.5 Requisitos Funcionales	24
2.3 Interfaces de usuario	28
2.4 Diseño del sistema.....	35
2.4.1 Cronograma entregables	36
2.4.2 Tarjetas CRC	37
2.4.3 Unified Modeling Language (UML)	40
2.4.3.1 Diagrama de despliegue:.....	40

2.5 Diseño de la Base de Datos.....	41
2.5.1 Modelo lógico.....	42
2.5.2 Modelo Físico.....	43
2.5.3 Diccionario de datos	44
2.6 Análisis de factibilidad	55
2.6.1 Factibilidad técnica:.....	55
2.6.1.1 Hardware	55
2.6.1.2 Software	56
2.6.2 Factibilidad económica:	56
2.6.2.1 Análisis costos-beneficios	57
CAPÍTULO 3	58
IMPLEMENTACIÓN Y PRUEBAS.....	58
3.1 Requerimientos de ejecución	58
3.1.1 Interfaces de hardware	58
3.1.2 Interfaces de software	58
3.1.3 Interfaces de comunicación.....	58
3.1.4 Interfaces de hardware del servidor.....	59
3.2 Seguridad	59
3.3 Mantenimiento	60
3.3.1 Manual de instalación	61
3.3.2 Manual de usuario	68
3.4 Pruebas.....	92
3.4.1 Pruebas de unidad.....	92
3.4.2 Pruebas de integración	95
3.4.3 Pruebas de seguridad	96
CONCLUSIONES	99
RECOMENDACIONES.....	100
LISTA DE REFERENCIAS	101
ANEXOS	104

ÍNDICE DE FIGURAS

Figura 1. (Gómez, 2014)	20
Figura 2. Descripción de los diferentes perfiles y sus accesos.....	22
Figura 3. Interfaz de inicio de sesión.....	28
Figura 4. Interfaz de inicio de sesión.....	29
Figura 5. Interfaz de inicio de sesión del aplicativo web.....	30
Figura 6. Interfaz de inicio de sesión / cambio de clave.....	30
Figura 7. Prototipo de interfaz de sesión autenticado.....	31
Figura 8. Carga de información estudiantes.....	32
Figura 10. Ingreso de las tutorías brindadas por parte de los docentes	33
Figura 11. Calificación de las tutorías por parte de los estudiantes	34
Figura 12. Diagrama de Despliegue	41
Figura 13. Modelo Lógico de la base de datos.....	42
Figura 14. Modelo Físico de la base de datos	43
Figura 15. Código Java – Código Encriptación MD5	59
Figura 16. Código Java – Implementación encriptación en la aplicación web	60
Figura 17. Encriptación de claves – Base de Datos.....	60
Figura 18. PostgreSQL – Creación de base de datos.....	61
Figura 19. PostgreSQL – Restauración de la base de datos.....	62
Figura 20. PostgreSQL – Base de datos (gestionaca).....	62
Figura 21. Glassfish – Interfaz gráfica.....	63
Figura 22. Configuración de Glassfish – Crear un Pool de Conexiones JDBC	63
Figura 23. Configuración de Glassfish – Recursos JDBC	64
Figura 24. Configuración de Glassfish – Verificación del Recurso JDBC creado .	64
Figura 25. Instalación de Hibernate.....	65
Figura 26. Instalación de Hibernate – Mensaje de confirmación.....	65
Figura 27. Archivos WAR y JAR en la ruta especificada.....	66
Figura 28. Archivos JAR – Deployar el archivo JAR	66
Figura 29. Archivos WAR – Deployar el archivo WAR	67
Figura 30. Visualización aplicativo	67
Figura 31. Autenticación.....	68

Figura 32. Horario	70
Figura 33. Cuestionario.....	71
Figura 34. Tutoría	72
Figura 35. Horario dentro de la tutoría	73
Figura 36. Estudiante no asignado.....	73
Figura 37. Historial estudiante	74
Figura 38. Reporte de horarios docentes.....	74
Figura 39. Imagen de reporte de calificación tutorías	75
Figura 40. Imagen de reporte de tutorías docente	75
Figura 41. Cuestionario.....	76
Figura 42. Reporte de tutorías estudiante.....	77
Figura 43. Página de inicio módulo administrador.....	78
Figura 44. Usuarios.....	79
Figura 45. Página encuestas	80
Figura 46. Página carga archivo	81
Figura 47. Página periodos.....	82
Figura 48. Página de carrera-facultad	83
Figura 49. Página de ubicación-espacio.....	84
Figura 50. Página de asignación tutoría	85
Figura 51. Reporte de la asignación tutoría.....	85
Figura 52. Página de la asignación menú.....	86
Figura 53. Imagen de filtro de horarios.....	87
Figura 54. Imagen de reporte de horarios	87
Figura 55. Reporte evaluaciones	88
Figura 56. Reporte de calificación tutorías	89
Figura 57. Imagen de la pantalla de filtros del reporte	90
Figura 58. Reporte de tutorías	90
Figura 59. Reporte de tutorías alumnos	91
Figura 60. Reporte de tutorías estudiantes	92
Figura 61. Prueba de unidad autenticación de usuarios.....	93
Figura 62. Prueba de unidad ingreso de una tutoría.	94
Figura 63. Prueba de unidad calificación de una tutoría.....	94
Figura 64. Interfaz ingreso de horarios	95
Figura 65. Interfaz de Tutorías	95

Figura 66. URL Conocido (inicio.xhtml) en el navegador	96
Figura 67. Sesión Caducada	96
Figura 68. Perfil de administrador	97
Figura 69. Perfil de estudiante	97
Figura 70. Perfil de docente	98
Figura 71. Perfil de Director de Carrera	98

ÍNDICE DE TABLAS

Tabla 1. Ventajas y desventajas de la arquitectura de tres capas	21
Tabla 2. Historia de usuario de seguridad y sesiones	24
Tabla 3. Historia de usuario gestión de usuarios	24
Tabla 4. Historia de usuario de gestión de perfiles.....	25
Tabla 5. Historia de usuario de módulo de horarios docente	26
Tabla 6. Historia de usuario de módulo de gestión tutoría.....	26
Tabla 7. Historia de usuario de módulo de gestión encuesta	27
Tabla 8. Historia de usuario de módulo de gestión estudiantes.....	27
Tabla 12. Cronograma de entregables de historias de usuario y tarjetas CRC.....	36
Tabla 13. Tarjeta CRC – Seguridad del Sistema	37
Tabla 14. Tarjeta CRC – Ingreso Usuario.....	37
Tabla 15. Tarjeta CRC - Perfiles	38
Tabla 16. Tarjeta CRC – Módulos.....	38
Tabla 17. Tarjeta CRC – Ingreso de Estudiantes.....	38
Tabla 18. Tarjeta CRC – Horarios Tutorías	39
Tabla 19. Tarjeta CRC – Registro de la Tutorías	39
Tabla 20. Tarjeta CRC – Calificación de la Tutoría por parte del Estudiantes	39
Nota. Tarjeta CRC – Calificación de la Tutoría por parte del Estudiantes	39
Tabla 21. Tarjeta CRC - Reportes.....	39
Tabla 22. Diccionario de datos de la tabla campus.....	44
Tabla 23. Diccionario de datos de la tabla modalidad	44
Tabla 24. Diccionario de datos de la tabla facultad.....	45
Tabla 25. Diccionario de datos de la tabla carrera.....	45
Tabla 26. Diccionario de datos de la tabla sede.....	46
Tabla 27. Diccionario de datos de la tabla distribución	46
Tabla 28. Diccionario de datos de la tabla lugar	47
Tabla 29. Diccionario de datos de la tabla espacio.....	47
Tabla 30. Diccionario de datos de la tabla ubicación	47
Tabla 31. Diccionario de datos de la tabla distlugar.....	48
Tabla 32. Diccionario de datos de la tabla materia.....	48
Tabla 33. Diccionario de datos de la tabla grupo	49

Tabla 34. Diccionario de datos de la tabla gestionestudiante.....	49
Tabla 35. Diccionario de datos de la tabla gestiontutoria	50
Tabla 36. Diccionario de datos de la tabla profesores	50
Tabla 37. Diccionario de datos de la tabla horario	51
Tabla 38. Diccionario de datos de la tabla pregunta.....	51
Tabla 39. Diccionario de datos de la tabla gestioncuestionario	52
Tabla 40. Diccionario de datos de la tabla opción.....	52
Tabla 41. Diccionario de datos de la tabla perfiles.....	53
Tabla 42. Diccionario de datos de la tabla perfilmenu	53
Tabla 43. Diccionario de datos de la tabla menú.....	54
Tabla 44. Diccionario de datos de la tabla gestionusuarios	54
Tabla 45. Software disponible	56
Tabla 46. Costo de desarrollo senior.....	56
Tabla 47. Costos de papelería y material de oficina	57

ÍNDICE DE ANEXOS

Anexo 1. Formato de la publicacion horarios tutoria	104
Anexo 2. Registro del tutor de asistencia a tutorías	105
Anexo 3. Registro del tutorado	106
Anexo 4. Formato del archivo a subir al sistema	107

RESUMEN

El proyecto de tesis que se ha desarrollado consiste en un aplicativo web que permite el registro y control de las tutorías académicas que brinda la Universidad Politécnica Salesiana sede Quito como parte de su proceso de mejoramiento en la calidad de la educación, mejorando el proceso de ingreso de la información se puede obtener de una manera más ágil indicadores que ayuden a la toma de decisiones de una manera oportuna que ayude a evitar que muchos estudiantes vean truncado su futuro profesional por falta de un apoyo académica adicional que la Universidad no ha podido brindarle ya que no contaba con la obtención de información de una manera inmediata.

El sistema presenta varios reportes de tipo gerenciales y estadísticos que se generan vial online y que contienen la información actualizada y confiable de cómo se está desarrollando este apoyo académico por parte de los docentes y estudiantes.

ABSTRACT

The thesis project that has developed is a web application that allows recording and monitoring of academic tutoring provided by the Salesian Polytechnic University Headquarters Quito as part of the process of improving the quality of education, improving the admissions process information can be obtained in a more agile indicators to help decision making in a timely manner to help prevent many students see truncated their professional future for lack of an additional academic support that the University has been unable to provide and that it did not obtain information in an immediate way.

The system has several types of management reports and statistics that are generated online and vial containing the date and reliable information on how you are developing this academic support from teachers and students.

INTRODUCCIÓN

La carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana no cuenta con un aplicativo web para el registro y control de la tutorías académicas por lo cual existe la necesidad de desarrollar un software que permita registrar todo el proceso de una tutoría que va desde el ingreso de los horarios de los docentes hasta la calificación del estudiante a la tutoría recibida, ya que de esta manera se obtendrán indicadores de gestión que permitan medir el desempeño que ha tenido tanto el docente como el estudiante en dicho apoyo académico y poder tomar las medidas necesarias si así lo amerita de una manera más rápida.

Este sistema fue diseñado en un ambiente web el cual utiliza el IDE Netbeans Java J2EE. Su estructura está basado en una arquitectura de tres capas: los servicios del sistema, la interfaz web y los datos. Para las interfaces del ambiente web se utilizaron componentes de Primefaces los cuales mejoran la experiencia del usuario en el manejo del aplicativo.

Este proyecto de tesis se ha desarrollado en cuatro capítulos:

El capítulo uno se basa en la importancia que tendrá la aplicación para la carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana, para ello se definió el problema y se planteó la solución, el cual fue diseñar un aplicativo que permita recopilar información necesaria de las tutorías académicas, también se describe la situación actual de la carrera con respecto al proceso de tutorías.

El capítulo dos se basa en la recolección de los requerimientos, con los que debe cumplir el aplicativo, para ello se utilizó el estándar IEEE-830 que permite obtener todas las necesidades que son requeridas para una óptima satisfacción del cliente, también se utiliza la Metodología XP, los modelos de interfaces y los diferentes entregables como son: historias de usuarios, tarjetas CRC, cronograma entre otros.

El capítulo tres se basa en los requerimientos de ejecución y la elaboración de pruebas al aplicativo, las cuáles permitieron verificar el correcto funcionamiento del software, las pruebas aplicadas fueron de seguridad, unidad e integración, se especifica también el manual de uso del aplicativo donde describe los campos que se tiene, el cómo llenarlos y sus diferentes restricciones.

El capítulo cuatro es la parte de los resultados obtenidos de la aplicación y las recomendaciones de futuras versiones del aplicativo de registro y control de tutorías académicas.

CAPÍTULO I

ESTUDIO DE LA SITUACIÓN ACTUAL DE LAS TUTORIAS ACADÉMICAS EN LA CARRERA DE INGENIERÍA DE SISTEMAS

1.1 Introducción

La educación tiene la misión de permitir a todos sin excepción, hacer fructificar su talento y su capacidad de creación, lo que implica que cada uno pueda responsabilizarse de sí mismo para el logro de su proyecto personal y su realización será una contribución esencial en la búsqueda de un mundo más viable y más justo (Delors, 1996).

Reconocer el papel protagónico que el estudiante ocupa en la educación superior, requiere desarrollar y potenciar los conocimientos y habilidades en torno a su incorporación a una sociedad en constante cambio, lo cual obliga a centrar la atención en su persona.

En este contexto, en el modelo educativo de la universidad el alumno juega un rol activo y responsable en la construcción de su propio proceso formativo, a través de un sistema curricular flexible y adaptado a sus necesidades educativas, proporcionando oportunidades de apreciar y aprovechar sus talentos al participar en actividades culturales, deportivas, de atención a su salud, eligiendo su carga horaria de acuerdo a las posibilidades institucionales, así como: para desarrollar la habilidad de comunicarse en un idioma extranjero, de participar en programas de servicio social, prácticas profesionales, estancias de aprendizaje, en ambientes laborales reales y en proyectos de investigación.

La carrera de Ingeniería en Sistemas de la Universidad Politécnica Salesiana no cuenta con un sistema informático para la administración y control de las tutorías académicas, que brinda como apoyo a los estudiantes de segundas y terceras matrículas que tienen por obligación el asistir a dichos apoyos académicos.

Uno de los propósitos del actual modelo educativo es; habilitar al docente para que tenga un perfil integral que lo haga un verdadero participante en la docencia, la investigación, la extensión y la gestión, actuando como un facilitador, guía o promotor de aprendizajes significativos, favoreciendo la movilidad, participando en el diseño curricular por competencias y brinde asesorías académicas y tutorías, abarcando aspectos actitudinales y humanísticos que trasciendan el ámbito universitario.

Bajo este modelo educativo, la universidad asume el reto de formar profesionistas competentes, comprometidos, preparados para responder a las necesidades de la sociedad de la información y del conocimiento, así como; socialmente responsables mediante experiencias de aprendizaje vinculadas con los sectores productivos y sociales; generando conocimientos útiles y relevantes para el desarrollo económico y social de la entidad.

Esta propuesta está orientada a propiciar la implementación de un programa en la que se deben abordar entre otros temas: las principales causas del rezago o abandono de los estudios, y/o las dificultades para su continuidad en la universidad, lo cual ayudará a promover el mejoramiento de la calidad de la educación la cual debe estar centrada en el estudiante. Este sistema estará diseñado en una plataforma web el cual utiliza el IDE Netbeans Java J2EE. La estructura de este proyecto estará basado en una arquitectura de tres capas: los servicios del sistema, la interfaz web y los datos. Para la capa de presentación se utilizaron componentes de Primefaces los cuales mejoran la experiencia del usuario al manejar dicha aplicación.

1.2 Problema

La carrera de Ingeniería en Sistemas de la Universidad Politécnica Salesiana no cuenta con un sistema informático para la administración y control de las tutorías académicas, que brinda como apoyo a los estudiantes de segundas y terceras matrículas. El proceso actual que realizan los docentes es llenar una hoja con el nombre del estudiante que asistió a la tutoría, el tema tratado y el registro de su firma.

Al final de cada semestre el docente debe entregar esta hoja a la Dirección de Carrera como evidencia de las tutorías impartidas. Este proceso es lento y no facilita la obtención de reportes, por lo que existe la necesidad de desarrollar un software de control de tutorías académicas, a través del cual se pueda registrar todo el proceso y seguimiento del docente con respecto al apoyo brindado y de esa manera alcanzar indicadores de gestión de forma inmediata.

Dichos indicadores permiten medir el desempeño que brindan las tutorías hacia los estudiantes y tomar decisiones para mejorar su calidad. Actualmente las tutorías asignadas a los docentes son de acuerdo al área de conocimiento que maneja el tutor, por lo cual el método que se maneja actualmente no permite tener una adecuada forma de toma de decisiones para el mejoramiento a tiempo de dicho apoyo académico.

1.3 Justificación

La Universidad Politécnica Salesiana, transita por un proceso de cambio y mejoramiento en su manejo de las tutorías académicas hacia los estudiantes que han reprobado las diferentes materias brindadas en el record académico, lo que hace necesario la construcción de un software que contribuya a mejorar la calidad de la gestión institucional impartida.

En la Universidad Politécnica Salesiana en la aplicación del sistema actual no permite obtener métricas que ayudaran a mejorar la calidad de las mismas de forma inmediata.

Contar con un sistema automatizado que permita realizar procesos académicos relacionados con la administración y control de las tutorías académicas, en forma ágil y eficiente, de tal manera que se pueda obtener información rápida y confiable, que permita a la Dirección de Carrera tomar decisiones sobre los indicadores generados.

La complejidad que existe en la actualidad para la acreditación de las carreras universitarias, han obligado a plantearse propuestas que estén de acuerdo con lo establecido por las entidades reguladoras de la Educación Superior como son, SENESCYT (Secretaría Nacional de Educación Superior, Ciencia y Tecnología), CEAACES (Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior), CES (Consejo de Educación Superior) y además con las leyes y reglamentos, como: la Ley Orgánica de Educación Superior, la Ley Nacional de Educación Superior, las cuales ayudan a la transparencia del sistema educativo, para que la formación impartida por dichas entidades sea del más alto nivel académico y permita contribuir al desarrollo humano.

Así que un sistema web para la administración y control de las tutorías académicas, constituye una forma indispensable de obtener una fuente de consulta ágil para la toma de decisiones dentro del esquema educativo de la carrera y para entidades externas que necesiten tener indicadores que validen el proceso brindado a los estudiantes.

De tal manera que se conseguirá un adecuado manejo de la información acerca de las tutorías académicas; lo cual ayudara a tomar decisiones para mejorar dicho proceso y brindar un apoyo confiable y que realmente sea una verdadera ayuda a los estudiantes.

1.4 Objetivos

1.4.1 Objetivo General

Desarrollo e implementación de un sistema para el registro y control de tutorías académicas, que permita realizar la administración y control de este proceso, y poder brindar un mejor apoyo académico a los estudiantes en riesgo de no poder continuar con su carrera, en un ambiente web, para la carrera de ingeniería en sistemas de la Universidad Politécnica Salesiana sede Quito campus sur.

1.4.2 Objetivos Específicos

- Identificar las necesidades y requerimientos que se manejan actualmente del sistema de administración y control de tutorías.
- Optimizar el proceso del registro y control de los estudiantes en situación de riesgo.
- Diseñar módulos e interfaces con la ayuda de la metodología programación extrema (XP) que permitan implementar el proceso optimizado para el control y seguimiento de tutorías.
- Mostrar los indicadores que se generarán en base al proceso optimizado de las tutorías, utilizando reportes de gestión y generales.

1.5 Alcance

La Universidad Politécnica Salesiana, transita por un mejoramiento en su proceso de gestión que brinda a los estudiantes que han reprobado las diferentes materias que se encuentran en el record académico y tienen la necesidad de recurrir a una tutoría académica, por lo cual hace necesaria la construcción de un software que permita el ingreso de información más ágil para obtener datos confiables y rápidos que ayuden a tomar decisiones que contribuyan a mejorar la calidad de dicha gestión institucional impartida.

Los actores que intervienen en el proceso de tutorías académicas son: docente, estudiante y director de carrera.

El proceso que realiza el actor docente le permite registrar sus horarios, los temas tratados en las tutorías, el ingreso de los estudiantes que asistieron, con lo cual al final del semestre

se puede generar un reporte con la información de todas las tutorías que ha brindado en ese período lectivo.

El proceso que realiza el actor estudiante permitirá realizar la calificación de la tutoría brindada y generar un reporte que indica a cuantas tutorías asistió dicho estudiante dentro del período lectivo, la calificación al tutor se lo realizará mediante la contestación de una encuesta que los estudiantes tendrán que llenar para completar el proceso de la tutoría, se le recordará al estudiante que debe calificar la tutoría mediante el envío de un e-mail a su correo institucional, las preguntas estarán relacionadas a saber si la tutoría recibida fue comprendida por el estudiante y cómo fue el desempeño del docente al brindarla.

El proceso que realiza el actor usuario final es la generación de reportes de tipo gerencial y estadísticos, dichos reportes gerenciales contemplaran información como el número de tutorados, temas tratados, horarios de docentes, número de tutorías brindadas, calificación de las tutorías por parte de los estudiantes.

En los reportes estadísticos se contemplará información de visual en barras del número de estudiantes que asisten a las tutorías, los temas más tratados, los profesores con más tutorías, los días con tutorías impartidas.

Cabe mencionar que el proceso de ingreso de la información lo realiza un administrador el cual tiene acceso a la carga de información y generación de reportes, lo cual le permitirá realizar cualquier cambio en el mismo.

Una debilidad del sistema es que no cuenta con la información completa de alumnos y docentes, ya que los estudiantes que recibirán las tutorías del período vigente serán cargados mediante un archivo en Excel el cual es enviado a la Dirección de Carrera al inicio de cada semestre, también el sistema no permitirá realizar ningún cálculo matemático y no tiene ninguna conexión con otros sistemas de la universidad.

1.6. Marco teórico

1.6.1. Tutorías Académicas

Según Calero Fernando en su libro titulado “Actividades de Tutoría” publicado en Madrid : ATEI, 2004, define a una tutoría como un proceso de acompañamiento académico hacia un estudiante o grupo de estudiantes por parte de docentes calificados designados como tutores, con fin el mejorar el rendimiento académico de dichos estudiantes.

Ser tutor requiere, como base un fuerte compromiso. Esta cualidad constituye la primera premisa sobre la que se construye el rol. Este compromiso tiene dos vertientes, tal como en el Documento de la AUNIES (2000) se explicita:

- Compromiso de adquirir la capacitación necesaria para la actividad tutorial.
- Compromiso de mantenerse informado sobre los aspectos institucionales y específicos del estudiante, esenciales para la actividad tutorial.

Las acciones que todo tutor realiza pueden reunirse en cuatro grupos:

- Contacto positivo con los alumnos a efectos de generar un clima de confianza que permita generar el vínculo deseado.
- Identificación de problemas justamente para contribuir con su solución.
- Toma de decisiones: debe tener la suficiente información y sentido de la oportunidad para hacerlo.
- Comunicación multidireccional.

Para cumplir con estas acciones, los tutores deben evidenciar las siguientes características personales:

Empatía – distancia: “Los tutores deberán generar las condiciones para una rápida y fluida comunicación con los tutorados pero sin perder de vista su rol, es decir, mantener la distancia necesaria que permita poner límites cuando sea adecuado.” (Calero, 2004, pág. 5)

Perseverancia: “Tenacidad para el logro de los objetivos y cumplimiento de las actividades.” (Calero, 2004, pág. 8)

Disponibilidad: “Deberán demostrar siempre voluntad para el encuentro y la comunicación más allá de que no sea posible en el exacto momento en el que el tutorando

lo propone, aquí adquiere relevancia la determinación del lugar y momentos en los que la intervención pueda concretarse.” (Calero, 2004, pág. 9)

Prudencia: “En relación con las afirmaciones y apreciaciones que haga. En ningún momento deberá realizar juicios de valor sobre las personas. En los casos en que deba emitir opinión, lo hará sobre la base de suficientes evidencias que la respalden.” (Calero, 2004, pág. 10)

Reserva: “Toda la información que reúna será comentada exclusivamente con los otros tutores, la Coordinadora del Programa, el Director de la Carrera y el Secretario Académico, en los casos en que resulte pertinente. Esta es una característica fundamental en su cometido que marca el encuadre ético en el que se tiene que desempeñar.” (Calero, 2004, pág. 12)

Conocimientos: “De aspectos reglamentarios y estructura curricular de la carrera.” (Calero, 2004, pág. 16)

1.6.2 Metodología XP (Extreme Programming)

Según LETELIER en su libro “Metodologías ágiles en el desarrollo de software”, la programación extrema o más conocida como: Extreme programming es una de las metodologías ágiles de desarrollo de software más exitosas de los últimos tiempos. Fue creada por Kent Beck en 1996, y tienen como objetivo aumentar la productividad a la hora de desarrollar programas, para lograr este objetivo se plantea 4 fases las cuales son: Planificación, Diseño, Codificación y Pruebas.

Fase 1: “La planificación es la primera actividad que se realiza en el proceso de desarrollo. Se crea un conjunto de historias de usuarios que describen la funcionalidad del software que se va a construir.” (Letelier, 2002, pág. 15)

Fase 2: “Diseño, dentro de esta metodología se sigue el principio de hacerlo todo simple, para que así, el diseño se pueda ir modificando a lo largo de todo el proceso de construcción según las exigencias del cliente.” (Letelier, 2002, pág. 155)

Fase 3: “Codificación, en esta fase los clientes y los desarrolladores del proyecto deben estar en comunicación para que los desarrolladores puedan codificar todo lo necesario para el proyecto que se requiere, en esta fase está incluido todo lo de codificación o programación por parte de los desarrolladores del proyecto.” (Letelier, 2002, pág. 200)

Fase 4: “Pruebas, uno de los pilares de la Metodología XP es el uso de pruebas para comprobar el funcionamiento de la aplicación que se ha implementado. En esta fase lo que se implementa son las pruebas de unidad, las cuales deben ser automatizadas para que puedan ejecutarse de manera fácil y rápida. De esta forma se puede modificar el código y asegurar el funcionamiento pese a los cambios producidos.” (Letelier, 2002, pág. 210)

Valores en Xp

Para poder implementar las prácticas que presenta XP hay que conocer cuáles son los valores principales que hacen a esta metodología. Estos valores son:

- Comunicación
- Simplicidad
- Feedback
- Coraje

Comunicación: “Uno de los valores más importantes en XP es la comunicación. La mala comunicación no surge por casualidad en un proyecto y puede aparecer en muchas circunstancias como por ejemplo: un programador le da malas noticias al gerente y este lo castiga, un cliente le dice al programador algo importante y este no le presta atención.” (Letelier, 2002, pág. 85)

Simplicidad: “La simplicidad es el segundo valor que se utiliza en esta metodología XP apuesta a realizar algo simple hoy y destinar un poco más de esfuerzo para realizar un cambio en el futuro, a realizar algo más complicado hoy y no utilizarlo nunca.” (Letelier, 2002, pág. 70)

Feedback: “Brindar un feedback correcto y preciso hace que se pueda mantener una buena comunicación y conocer el estado actual del proyecto.” (Letelier, 2002)

Coraje: “El coraje también es poder realizar cambios cuando algo no funciona del todo bien, diseñar e implementar solo lo necesario para el presente, pedir ayuda o reducir el alcance de una entrega si el tiempo no alcanza. Si no hay coraje en un proyecto no se puede clasificar como extremo y es necesario que los otros tres valores estén presentes.” (Letelier, 2002, pág. 30)

Principios de Xp

Las cuatro fases mencionadas anteriormente nos brindan un estándar para obtener buenos resultados. Sin embargo, los valores son muy vagos a la hora de ayudarnos a decidir que prácticas utilizar. Para ello se necesita destilar estos valores en principios que puedan ser utilizados.

A continuación se detallan los principios más importantes de XP:

- Rápida retroalimentación
- Asumir la simplicidad
- Cambios incrementales
- Aceptar el cambio
- Trabajo de calidad

Rápida retroalimentación: “En la práctica el tiempo transcurrido entre una acción y su feedback es crítico. Tener una rápida retroalimentación nos permite interpretarla, aprender de ella y poner en práctica lo asimilado lo antes posible.” (Letelier, 2002, pág. 255)

Asumir la simplicidad: “Este es uno de los principios más difíciles de llevar a la práctica. Casi siempre se planifica para el futuro y se diseña para poder rehusar. En lugar de esto XP dice que hay que hacer un buen trabajo para las necesidades actuales y confiar en nuestra habilidad para solucionar problemas futuros.” (Letelier, 2002, pág. 256)

Cambios incrementales: “Realizar grandes cambios en una sola oportunidad no es una buena solución. Cada problema debe ser resuelto con una serie de cambios pequeños para poder atacar dicho problema mucho más en profundidad.” (Letelier, 2002, pág. 257)

Aceptar el cambio: “En XP el cambio es asimilado como algo habitual e inevitable. La mejor estrategia es aquella que preserva la mayor cantidad de opciones mientras resuelve los problemas más precisos.” (Letelier, 2002, pág. 258)

Trabajo de calidad: “Uno de los objetivos más importantes en XP es realizar un producto de buena calidad. Si cada integrante realiza su trabajo de la mejor manera posible se puede asegurar la calidad del producto.” (Letelier, 2002, pág. 259)

Principales Conceptos

Story Cards: “Las story cards sirven para registrar los requerimientos de los clientes y son utilizadas para poder realizar la estimación de cada una de las iteraciones durante la fase de planificación. Las story cards son escritas por los clientes en base a lo que se estima es necesario para el sistema. Están escritas en un formato de oraciones en la terminología del cliente, sin necesidad de sintaxis técnicas.” (Letelier, 2002, pág. 399)

Iteración: “Consta de un período de una a dos semanas en las cuales el cliente selecciona las historias en ser desarrolladas. Luego de ser implementadas este cliente corre sus test funcionales para ver si la iteración puede terminar de manera exitosa.” (Letelier, 2002, pág. 400)

Refactoring: “Otro concepto muy importante es el refactoring. Consiste en realizar cambios al sistema sin modificar la funcionalidad del mismo para poder hacer el sistema más simple, para aumentar la eficiencia o para que el código sea mucho más entendible. Sea cual sea el objetivo del refactoring, no hay que olvidar que en XP el código es la documentación más importante del proyecto y por ende debe estar bien desarrollado.” (Letelier, 2002, pág. 402)

Release: “Un release puede ser definido como un conjunto de funcionalidad que es integrada para realizar un programa ejecutable. La idea de cada release es poder tener un producto intermedio al final de cada iteración en la cual el cliente pueda testear la funcionalidad pedida. Con esto los clientes pueden, además, ver el avance del proyecto y poder realizar comentarios a los programadores y no esperar hasta el final del mismo cuando esté todo integrado.” (Letelier, 2002, pág. 402)

Test de aceptación: “Los test de aceptación representan algún tipo de resultado por parte del sistema. Los clientes son los responsables de verificar la exactitud de estos test y de revisar los resultados para poder así priorizar los test que fracasaron. También son utilizados como test regresivos antes de entrar a la fase de producción.” (Letelier, 2002, pág. 403)

Test unitario: “Los testing unitarios son tan importantes como los test de aceptación. Son realizados desde el punto de vista del programador y sirven, además de testear el código, para poder realizar el refactoring del mismo. Cada programador, antes de comenzar a programar, debe preparar los test unitarios. Esto hace que dichos test estén

preparados para ser corridos durante la codificación y además, hace que al programador le surjan dudas y puedan ser solventadas con el cliente antes de empezar con la codificación.” (Letelier, 2002, pág. 403)

1.6.3 Arquitectura de tres capas

Según Gómez J. señala que el “diseño de sistemas informáticos actuales se suele usar las arquitecturas multinivel o programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables” (Gómez, 2014, pág. 58)

Capa de presentación: “Hace referencia a la interacción entre el usuario y el software. La principal responsabilidad es mostrar información al usuario.” (Gómez, 2014, pág. 58)

Para la visualización de éste es necesario hardware (PC, laptops, tablets, etc.) y un navegador web (Firefox, Explorer, Chrome).

Capa de lógica del negocio: “Esta capa contiene la funcionalidad que implementa la aplicación. Controla la ejecución de la capa de acceso a datos y servicios externos. Se puede diseñar la lógica de la capa de negocios para uso directo por parte de componentes de presentación o su encapsulamiento como servicio y llamada a través de una interfaz de servicios que coordina la conversación con los clientes del servicio o invoca cualquier flujo o componente de negocio” (Gómez, 2014, pág. 59)

Capa de datos: “Esta incluye la base de datos que contiene todos los datos de la organización y ésta se encapsula desde los usuarios finales. Cualquier número de 24 servidores de base de datos puede colocarse en esta capa, dependiendo del volumen de transacciones y de la cantidad de datos.” (Gómez, 2014, pág. 60)

1.7 Situación Actual

La Universidad Politécnica Salesiana actualmente en su proceso de tutorías académicas es guiada mediante un manual llamado “Manual de Tutores” que es enviado a los docentes a su correo institucional el cual es elaborado por el Físico Franklin Erazo docente de la Universidad Politécnica Salesiana y que contiene la siguiente información:

Manual para tutores

Este manual tiene por objetivo constituirse en material de primera mano para que tanto los tutores como los estudiantes-tutorados puedan desenvolverse de manera correcta y eficiente en la actividad llamada tutorías. Pretende facilitar la consulta rápida del tutor. Por favor léalo con detenimiento.

Tutoría

La tutoría u horas de consultas, es una actividad académica que consiste en otorgar a los estudiantes en riesgo de deserción (aquellos que están con 2da y 3ra matrícula), la oportunidad de recibir asistencia académica en falencias específicas. Con esto, la carrera pretende por un lado; reducir el nivel de pérdida de éstos estudiantes, y por otro, cumplir con la Ley de Educación Superior que demanda la existencia de éste espacio de consulta para los estudiantes.

Operatividad de las tutorías

Para esta actividad se sugiere utilizar el concepto de tutoría individualizada o personalizada, (atención a un estudiante o un grupo pequeño de ellos), la cual es una sesión de asesoría académica que dura un tiempo aproximado de 15 minutos por estudiante. Allí, el tutor absuelve dudas específicas que trae cada estudiante.

Evaluación de las tutorías

Para los tutores:

Cada tutor deberá registrar la asistencia a las tutorías en una hoja especial llamada “REGISTRO DEL TUTOR” (ver anexo 1, cada tutor debe imprimir esta hoja y sacar las copias que crea necesarias). La frecuencia normal de asistencia de los estudiantes es de 4 estudiantes por cada hora de tutoría. Por ejemplo: si un tutor tiene asignadas 4 horas semanales de tutorías en la materia de Métodos Numéricos, al finalizar el semestre, deberá reportar un total de registro de asistencia de por lo menos 100 tutorías (se cuentan por igual tanto en los casos cuando un estudiante hace una sencilla pregunta que es absuelta muy rápidamente por el tutor, como también cada vez que un estudiante repite la asistencia a tutorías, es decir, asiste regularmente). En general las tutorías se ofrecen a cualquier estudiante matriculado en la carrera de sistemas, pero en especial a quienes están en riesgo (para estos estudiantes la asistencia a tutorías es obligatoria).

Al finalizar el semestre (el plazo vence el último día de clases normales, antes de los exámenes del 2do parcial), cada tutor debe enviar y entregar un informe, donde adjunte el resto de registros de asistencia de los estudiantes.

Para los estudiantes en riesgo:

El tutor debe entregar una hoja llamada “Registro del tutorado” (ver anexo 2), a cada estudiante en riesgo. Esto lo debe hacer durante las primeras clases. Así mismo debe comunicar a éstos estudiantes la obligatoriedad de la asistencia a tutorías. Esta hoja debe ser firmada por el tutor una vez por cada tutoría y debe ser conservada por cada estudiante. Al finalizar el semestre, el estudiante debe presentar a su profesor-tutor, la hoja con por lo menos 10 registros de asistencia. Se sugiere al tutor considere el cumplimiento de este requerimiento por parte de cada estudiante en riesgo, haciéndole merecedor a 2 puntos (sobre 100), si así lo requiriese para aprobar el semestre.

Actividad compensatoria a las tutorías

Como no puede preverse el nivel de asistencia de los estudiantes a las tutorías, se establece una actividad llamada “Compensatoria”, consistente en la elaboración de un folleto didáctico de un tema escogido por el profesor (por ejemplo; el desarrollo de un capítulo del programa de la materia, con ejemplos y ejercicios resueltos). Esta actividad la deben realizar los tutores quienes pese haber informado y convocado a los estudiantes, éstos no asisten incumpliendo la normativa de la carrera. Como un parámetro aproximado se establece que el folleto debe tener una extensión de una página escrita en Word, letra tamaño 12, a doble espacio, con márgenes normales, por cada hora que transcurre mientras ningún estudiante asiste a las tutorías. Por ejemplo; si durante el semestre un docente-tutor tuvo 4 horas de inactividad por inasistencia de los estudiantes, entonces el folleto debe tener una extensión de 4 páginas y podría ser un par de exámenes resueltos o un formulario. Todos estos folletos que realizan los tutores deberán ser enviados como plazo máximo el último día de clases) vía e-mail a su jefe de departamento en archivo digital. El Consejo de Carrera buscará la manera de que estos trabajos sean editados, publicados y puestos a disposición de los estudiantes, incrementando por un lado el nivel de producción de material didáctico realizado por la carrera y por otro la carrera reconocerá a sus autores éstos aportes en sus hojas de vida y curricular personal.

Observaciones adicionales

1) Cada tutor debe informar durante las primeras clases del semestre a los estudiantes; el horario y lugar a realizarse las tutorías. Cada tutor debe trabajar en su escritorio asignado (en caso que algún tutor no tuviera asignado todavía un escritorio, por favor ubicarse en un aula cualquiera desocupada del 2do piso del bloque B. Si esta estuviera cerrada, por favor pedir a guardianía o a limpieza que se abra el aula).

2) Un mismo estudiante se registrará máximo una sola vez por día.

CAPÍTULO 2

ANÁLISIS Y DISEÑO DEL SISTEMA

2.1 Especificación del sistema de software

2.1.1 Introducción

Para realizar el levantamiento de los requerimientos de software se utilizará el estándar IEEE830, la cual se elaboró en el año de 1998 y permite recolectar de manera clara y específica todas las necesidades que debe cumplir la aplicación, entre las principales características que tiene esta norma para que el cliente puede especificar en los requerimientos son: una breve descripción general del producto que se desea, la funcionalidad con la que debe cumplir el software, las posibles restricciones que tendrá y cómo se validara el manejo de perfiles de usuarios entre otros.

Objetivos:

Los objetivos que se deben establecer en función del contexto de desarrollo de software se describen a continuación:

- Un cliente describa claramente lo que quiere
- Un proveedor entienda claramente lo que el cliente quiere
- Se establezcan bases para un contrato de desarrollo (o de compra-venta)
- Se reduzca el esfuerzo de análisis, diseño, y programación (evitando re-trabajos)

Consideraciones para una buena ERS

- Naturaleza de la ERS
- Ambiente de la ERS
- Características de una buena ERS
- Preparación conjunta de la ERS
- Evolución de la ERS
- Prototipos
- Diseño en la ERS
- Requisitos del proyecto en la ERS

Características de una buena ERS

Una buena ERS debe ser:

- Correcta
- Inequívoca
- Completa
- Con todos los requisitos relacionados con funcionalidad, rendimiento, restricciones de diseño, atributos e interfaces externas.
- Respuestas a todas las posibles entradas (válidas e inválidas)
- Con todas las etiquetas y referencias a figuras, tablas, diagramas en la ERS
- Definición de las unidades de medida.
- Consistente
- Organizada por orden de importancia y/o estabilidad Esencial, condicionada a u opcional – Con/sin cambios
- Comprobable
- Modificable
- Trazable

2.2 Descripción general

La aplicación que se pretende desarrollar es un software de administración y control de tutorías académicas para la carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana Sede Quito Campus Sur, el cual permitirá al Director de Carrera manejar indicadores relacionados con las tutorías impartidas por los docentes de la carrera de Ingeniería en Sistemas. Mientras que al docente se le permitirá el ingreso de los horarios en los cuales impartirán las tutorías, registrar el tema de la tutoría y a los alumnos que recibieron dicho apoyo académico, para luego generar un reporte de su gestión como tutor, en el caso de ser un alumno estará obligado a calificar la tutoría recibida, podrá generar un reporte de su asistencia. El administrador estará encargado de alimentar la base de estudiantes por período lectivo, dar acceso a los diferentes perfiles, cargar la base de docentes y la base de materias.

Los reportes que generará el aplicativo son de tipo generales y de gestión en los cuales se indicará la cantidad y la calidad de la tutoría que ha brindado un docente en un período lectivo.

Los reportes que se pretende realizar con este aplicativo son los siguientes:

- Reporte de horarios docentes
- Reporte de la asistencia por estudiante a la tutoría.
- Reportes del número de tutorías brindadas por el docente
- Reporte de la calificación de la tutoría al docente.

2.2.1 Perspectiva del producto

Para descartar que exista una aplicación que permita generar la información necesaria se realizó una investigación previa para conocer de temas o proyectos similares al que se desarrollará, y no se encontró ningún sistema que permita generar la información que se necesita acerca de las tutorías académicas y que el sistema a desarrollar le permitirá a la Directora de Carrera realizar.

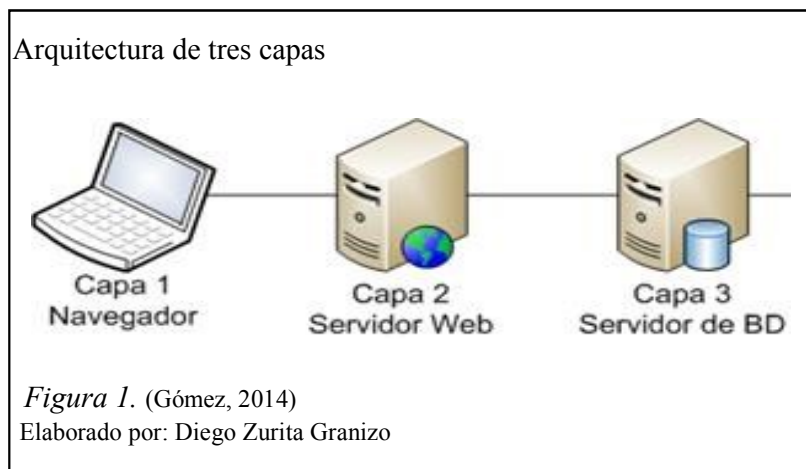
El sistema que se encontró están relacionados a las tutorías vía online que se lo desarrollo en México:

- María Cristina Dávila Avendaño implementación de tutoría en línea para alumnos de la Facultad de Ingenierías de la Universidad del Valle de Atemajac en un entorno virtual de aprendizaje, 2013, México.

Después de haber culminado con la investigación y no haber encontrado otro tema similar al que se desarrolla, se puede indicar que el aplicativo que se está diseñando es único en su clase ya que va a permitir a la carrera de Ingeniería en Sistemas conseguir toda la información requerida acerca de las tutorías académicas de una manera más rápida que brindan los docentes a los estudiantes en riesgo de deserción.

2.2.2 Arquitectura de tres capas

“El diseño de los sistemas informáticos actualmente se utiliza las arquitecturas multinivel o programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables.” (Universidad de Guayaquil, 2008)



Capa de presentación: hace referencia a la interacción entre el usuario y el software. La principal responsabilidad es mostrar información al usuario. Para la visualización de éste es necesario hardware (PC, laptops, tablets, etc.) y un navegador web (Firefox, Explorer, Chrome).

Capa de lógica del negocio: “Esta capa contiene la funcionalidad que implementa la aplicación. Controla la ejecución de la capa de acceso a datos y servicios externos. Se puede diseñar la lógica de la capa de negocios para uso directo por parte de componentes de presentación o su encapsulamiento como servicio y llamada a través de una interfaz de servicios que coordina la conversación con los clientes del servicio o invoca cualquier flujo o componente de negocio.” (Freire, 2002)

DAO: “Las aplicaciones de negocios necesitan tener acceso a los datos casi siempre de bases de datos relacionales o de objetos y la plataforma Java” (Dao, 2014)

Entidad: “Es un objeto de dominio de persistencia. Normalmente, una entidad representa una tabla en el modelo de datos relacionales y cada instancia de esta entidad corresponde a un registro en esa tabla.” (Jpa, 2009)

Servicios: “Capa transaccional, la cual permite la interacción de la lógica de negocio con cualquier capa de presentación.” (Universidad de Talca, 2013)

Archivos de configuración (pesist.xml): “El fichero de configuración persistence.xml permite especificar las bases de datos accesibles a una aplicación y las entidades que utiliza.” (Universidad de Talca, 2013)

Capa de datos: “Esta incluye la base de datos que contiene todos los datos de la organización y esta se encapsula desde los usuarios finales. Cualquier número de servidores de base de datos puede colocarse en esta capa, dependiendo del volumen de transacciones y de la cantidad de datos.” (Easten software systems pvt. ltd, 2006)

Ventajas y desventajas de la arquitectura de tres capas

Tabla 1. *Ventajas y desventajas de la arquitectura de tres capas*

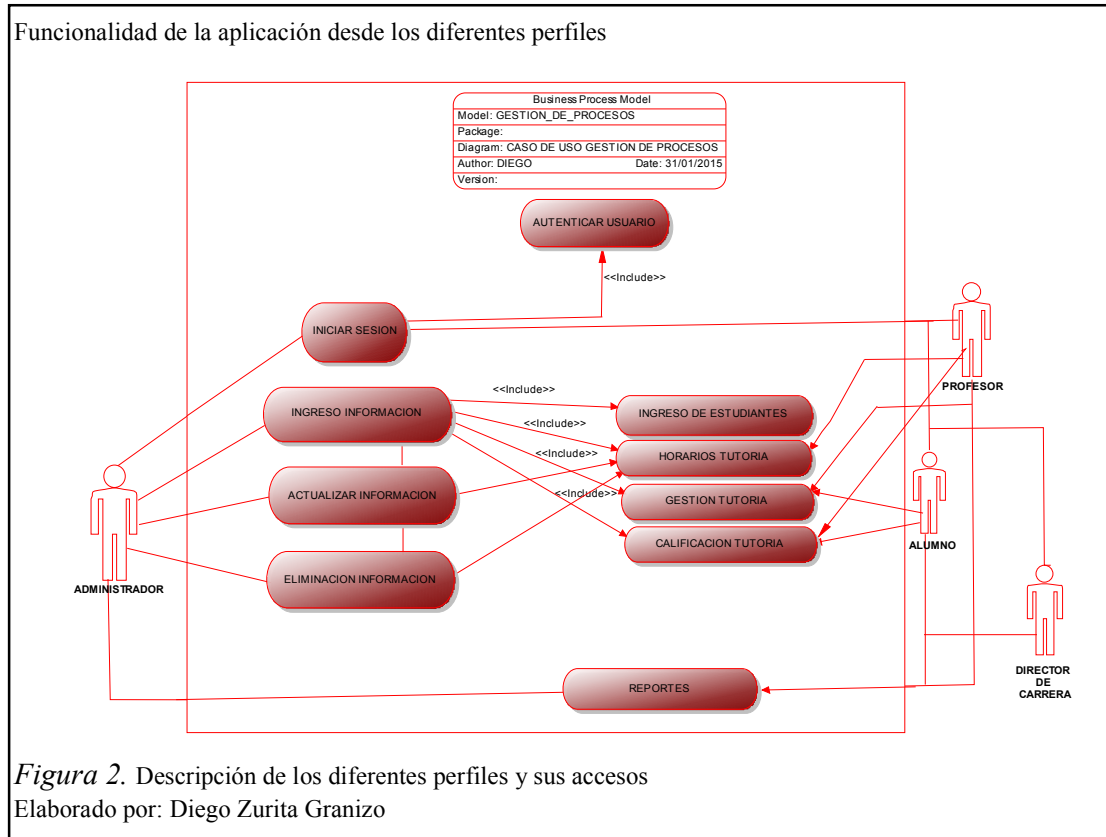
Ventajas	Desventajas
<ul style="list-style-type: none"> • Con la arquitectura de tres capas, la interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos. Por lo tanto, esa estructura de los datos puede ser modificada sin cambiar la interfaz del usuario en la PC. • El código de la capa intermedia puede ser reutilizado por múltiples aplicaciones si está diseñado en formato modular. • La separación de roles en tres capas, hace más fácil reemplazar o modificar una capa sin afectar a los módulos restantes. 	<ul style="list-style-type: none"> • Los ambientes de tres capas pueden incrementar el tráfico en la red y requiere más balance de carga u tolerancia a las fallas. • Los exploradores actuales no son todos iguales. • La estandarización entre diferentes proveedores ha sido lenta en desarrollarse. Muchas organizaciones son forzadas a escoger uno en lugar de otro, mientras que cada uno ofrece sus propias y distintas ventajas.

Nota. Ventajas y desventajas de la arquitectura de tres capas (Instituto Tecnológico Veracruz, 2014)

Elaborado por: Diego Zurita Granizo

2.2.3 Función del producto

En el siguiente caso de uso se muestra la funcionalidad que tiene la aplicación de una forma muy general en la cual nos permite apreciar el manejo de los perfiles que se utilizarán, con sus respectivas opciones de acceso que poseen.



Usuario Administrador: es el encargado de realizar todos los mantenimientos e ingresos de datos de los estudiantes en riesgo de deserción en el aplicativo, dar acceso a los diferentes perfiles y generar reportes.

Usuario Profesor: es el encargado de realizar el ingreso de las tutorías impartidas a los estudiantes en riesgo de deserción.

Usuario Estudiante: es el encargado de realizar la calificación de las tutorías recibidas por parte de un docente y poder generar su reporte de asistencia a las mismas.

Usuario Director de Carrera: solo se le permitirá la generación de los reportes.

2.2.4 Limitaciones generales

Al igual que otros sistemas, este ambiente web posee también sus limitaciones puesto que no contiene toda la información de los docentes como por ejemplo:

- La información de los estudiantes se la carga a través de un archivo en Excel al inicio de cada semestre.
- Los docentes y materias se alimenta a través del archivo que contiene a los estudiantes que van a recibir las tutoría.
- No se integra con otros sistemas existentes en la universidad
- No muestra información de los horarios de los docentes en las materias asignadas en ese período.

Esta aplicación no realiza estos procesos debido a que el ingreso se lo realiza por medio de un archivo en Excel que es enviado al inicio de cada semestre con la información de los estudiantes en riesgo de deserción con las respectivas materias y docentes.

La aplicación se encuentra orientada a la gestión de las tutorías académicas en cuanto a lo que se refiere: ingreso de estudiantes, ingreso de horarios de las tutorías por parte de los docentes, ingreso de estudiantes que recibieron la tutoría, calificación de la tutoría por parte del estudiante, generación de reportes.

2.2.5 Requisitos Funcionales

Los requisitos funcionales nos ayudan a obtener a grandes rasgos las funcionalidades del futuro sistema que se pretende desarrollar.

Tabla 2. *Historia de usuario de seguridad y sesiones*

Historia de usuario	
Número: 1	Usuario: Washington Padilla
Nombre historia: Seguridad y Sesiones	
Programadores responsables: Diego Zurita Granizo	
Descripción: Lo que se debe realizar primero es la autenticación para poder ingresar al Sistema de Ambiente Web para insertar, actualizar, eliminar y consultar información dependiendo del perfil de usuario que tenga.	
Observaciones: Solo usuarios autenticados pueden consultar datos en el ambiente web.	

Nota. Descripción de las seguridades

Elaborado por: Diego Zurita Granizo

Tabla 3. *Historia de usuario gestión de usuarios*

Historia de usuario	
Número: 2	Usuario: Washington Padilla
Nombre historia: Gestión de Usuarios	
Programadores responsables: Diego Zurita	
Descripción: Para el manejo de usuarios el único que puede realizarlo es el usuario administrador debido a que él tiene acceso total.	
Observaciones: El usuario Administrador puede crear nuevos usuarios, actualizar y eliminarlos.	

Nota. Manejo de los usuarios

Elaborado por: Diego Zurita Granizo

Tabla 4. *Historia de usuario de gestión de perfiles*

Historia de usuario	
Número: 3	Usuario: Washington Padilla
Nombre historia: Gestión de Perfiles	
Programadores responsables: Diego Zurita Granizo	
Descripción: Se necesita contemplar cuatro tipos de perfiles los cuales son administrador, visitante, docente y estudiante. Los cuales tendrán acceso a los diferentes módulos que manejará el Sistema.	
Observaciones: Administrador: Acceso Total (Creación de nuevos usuarios, actualizar y eliminar información de todos los módulos, visualización de reportes) Visitante: Visualización de Reportes. Estudiante: Acceso a calificar las tutorías recibidas y a la visualización de reportes Docente: Ingreso de sus horarios para las tutorías, ingreso de Datos de las Tutorías por estudiante, y acceso a reportes.	

Nota. Manejo de perfiles del aplicativo

Elaborado por: Diego Zurita Granizo

Tabla 5. *Historia de usuario de módulo de horarios docente*

Historia de usuario	
Número: 4	Usuario: Washington Padilla
Nombre historia: Gestión Horarios Docente	
Programadores responsables: Diego Zurita Granizo	
Descripción: Se necesita poder ingresar la información de los horarios de los docentes que brindaran la tutorial para esto se necesita tener los días que brindará la tutoría, hora el lugar donde se realiza la tutoría, la materia y el grupo.	
Observaciones: A este módulo los únicos usuarios que pueden ingresar son los que tengan perfil de docente.	

Nota. Gestión de los horarios del docente para la tutoría

Elaborado por: Diego Zurita Granizo

Tabla 6. *Historia de usuario de módulo de gestión tutoría*

Historia de usuario	
Número: 5	Usuario: Washington Padilla
Nombre historia: Gestión Tutoría	
Programadores responsables: Diego Zurita Granizo	
Descripción: Se necesita poder ingresar la información de la tutoría que ha brindado un docente donde se requiere la materia el grupo los estudiantes el tema tratado y la duración de la tutoría y el horario de la tutoría, las acciones que se pueden hacer es ingresar.	
Observaciones: A este módulo los únicos usuarios que pueden ingresar son los que tengan perfil de docente.	

Nota. Gestión de la tutoría impartida

Elaborado por: Diego Zurita Granizo

Tabla 7. *Historia de usuario de módulo de gestión encuesta*

Historia de usuario	
Número: 6	Usuario: Washington Padilla
Nombre historia: Calificación Tutoría	
Programadores responsables: Diego Zurita Granizo	
Descripción: Se necesita poder calificar la tutoría que ha impartido un docente con la contestación de 5 preguntas.	
Observaciones: A este módulo los únicos usuarios que pueden ingresar son los que tengan perfil de estudiante.	

Nota. Calificación de la tutoría impartida

Elaborado por: Diego Zurita Granizo

Tabla 8. *Historia de usuario de módulo de gestión estudiantes*

Historia de usuario	
Número: 7	Usuario: Diego Zurita
Nombre historia: Gestión Estudiantes	
Programadores responsables: Diego Zurita Granizo	
Descripción: Se necesita poder ingresar a los estudiantes que recibirán la tutoría mediante un archivo en Excel que contendrá la siguiente información: estudiante, cédula, materia, número de veces que repite la materia, carrera, grupo y la validación. Las acciones que se podrán realizar es el ingreso de datos.	
Observaciones: A este módulo los únicos usuarios que pueden ingresar son los que tengan perfil de administrador.	

Nota. Gestión de los estudiantes que recibirán la tutoría

Elaborado por: Diego Zurita Granizo

2.3 Interfaces de usuario

El sistema que se desarrolla presentan cuatro interfaces principales donde se realiza el ingreso de toda la información acerca de los estudiantes que están en riesgo, los horarios de los docentes para brindar la tutoría, las tutoría brindadas a los estudiantes y la calificación de dichas tutorías por parte de los estudiantes para con esta información poder generar los reportes necesarios vía online de manera inmediata.

Para las interfaces de esta aplicación se ha definido que todas contengan una cabecera donde se deberá colocar el logo de la Universidad Politécnica Salesiana y el nombre de la aplicación, una sección media donde se colocara la navegabilidad del sistema mediante un menú de opciones y un pie de página donde se colocara la información del desarrollador del aplicativo.

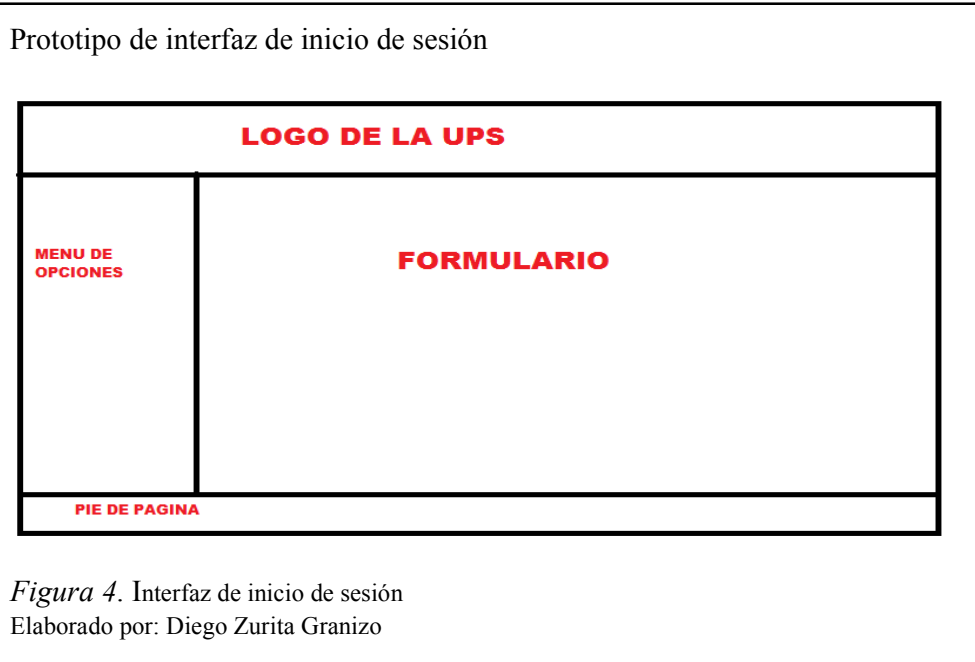
Para la interfaz de inicio de sesión la estructura será la siguiente:

Prototipo de interfaz de inicio de sesión

LOGO DE LA UPS	
FORMULARIO	BREVE DESCRIPCION DEL APLICATIVO
PIE DE PAGINA	

Figura 3. Interfaz de inicio de sesión
Elaborado por: Diego Zurita Granizo

Para las demás interfaces la estructura de su diseño será la siguiente:



Interfaz de inicio de sesión: en la parte superior de la misma se encuentra el logo de la universidad y el nombre de la aplicación. En la parte central izquierda se encuentra una pequeña descripción de la aplicación. En la parte central derecha se encuentra un panel de autenticación, el cual contiene a una caja de texto que pedirá el usuario y otra que pedirá la clave y un botón que permitirá autenticarse y otro que permitirá realizar el cambio de clave si así lo deseara.

Prototipo de interfaz de inicio de sesión

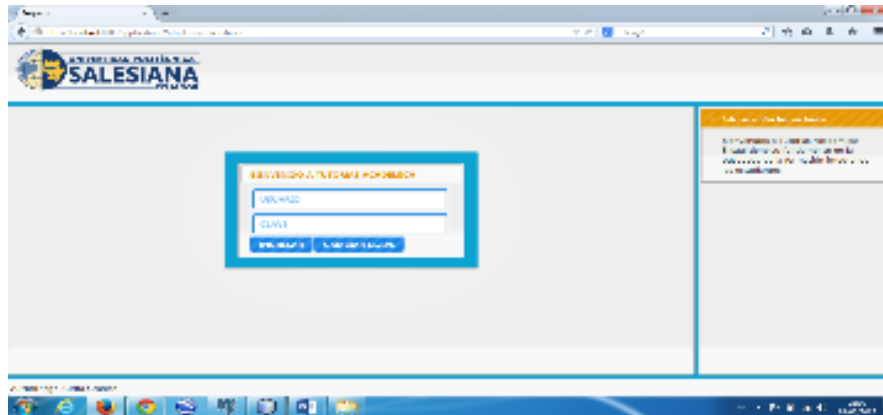


Figura 5. Interfaz de inicio de sesión del aplicativo web
Elaborado por: Diego Zurita Granizo

Prototipo de interfaz de inicio de sesión / cambio de clave



Figura 6. Interfaz de inicio de sesión / cambio de clave
Elaborado por: Diego Zurita Granizo

[illegible]

La información que se presentara en pantalla es la del estudiante y los datos a mostrar son los siguientes: Estudiante, Cédula, Materia, Num_Repite, Carrera, Grupo.

Prototipo de carga de información estudiantes

Figura 8. Carga de información estudiantes

Elaborado por: Diego Zurita Granizo

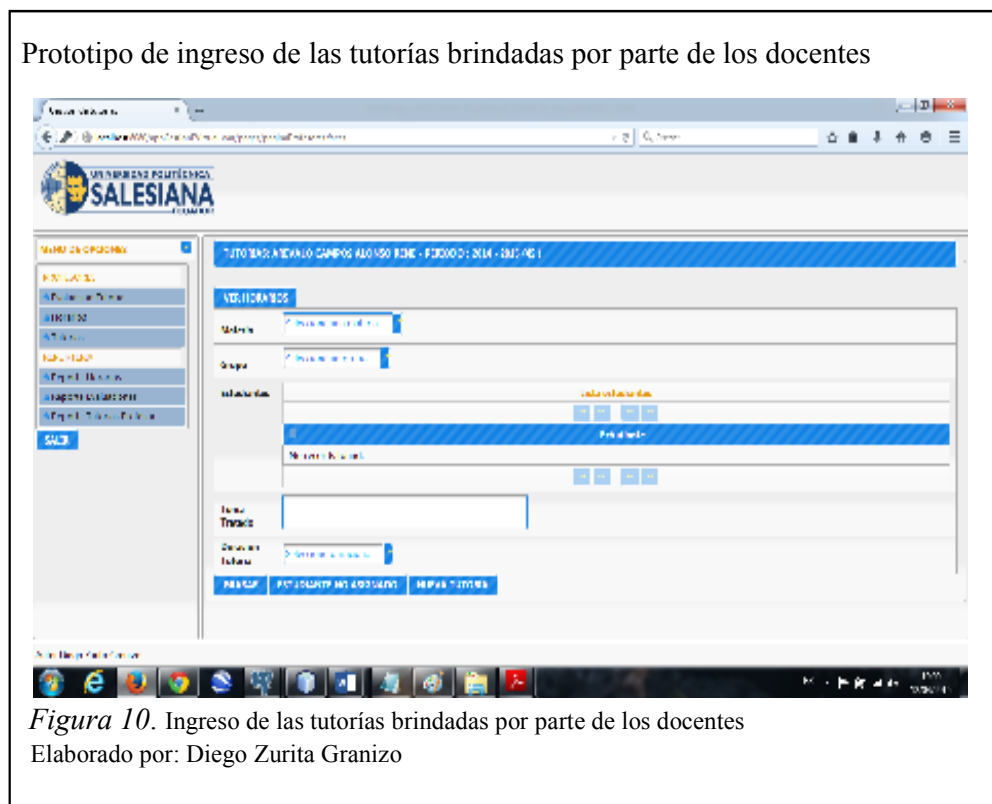
Interfaz de carga de horarios docentes: para esta interfaz de registro de nueva información acerca de los horarios de los docentes para brindar la tutoría, el cual solicitará el ingreso de varios campos dependiendo de la información que solicite el formulario.

Prototipo de carga de horarios docentes

Figura 9. Carga de horarios docentes

Elaborado por: Diego Zurita Granizo

Interfaz de ingreso de la tutoría: para esta interfaz de registro de nueva información acerca de las tutorías que los docentes brindan a los estudiantes, en la parte superior aparecerá el logo de la Universidad Politécnica Salesiana, en la parte izquierda estará el menú de navegación y el formulario aparecerá en la parte central, el cual solicitará el ingreso de varios campos dependiendo la información que solicite el formulario, luego del ingreso de dicha información se le enviará un e-mail al estudiante informándole que debe calificar la tutoría.



Interfaz para la calificación de la tutoría por parte de los estudiantes: para esta interfaz de registro de nueva información acerca de la calificación que realiza el estudiante hacia la tutoría que los docentes han brindado, en la parte superior aparecerá el logo de la Universidad Politécnica Salesiana, en la parte izquierda estará el menú de navegación y el formulario aparecerá en la parte central, el cual solicitará el ingreso de la contestación de las preguntas que hacen referencia a la manera que tuvo el docente en brindar dicho apoyo académico.

Las preguntas que se han configurado en el aplicativo son las siguientes:

- La cordialidad y capacidad del tutor logra crear un clima de confianza para que puedas exponer tus problemas
- El tutor posee formación profesional en tu especialidad
- Consideras que el tutor posee dominio de métodos pedagógicos para la atención individualizada o grupal, según sea el caso
- El tutor te trata con respeto y atención
- El tutor tiene interés en detectar tus principales dificultades y realizar las acciones pertinentes para resolverlas
- EL docente mostro dominio del tema tratado en la tutoría.
- Tu integración a la universidad ha mejorado con el programa de tutoría
- El tutor te canaliza a las instancias adecuadas cuando tienes algún problema que rebasa su área de acción.
- Consideras que tu participación en el programa de tutoría ha mejorado tu desempeño académico.
- Consideras que el programa de tutoría es satisfactorio
- El tutor tiene capacidad para resolver tus dudas académicas
- El tutor tiene interés en orientarte en metodología y técnicas de estudio

Prototipo para la calificación de las tutorías por parte de los estudiantes

Elaborado por: Diego Zurita Granizo

2.4 Diseño del sistema

Luego del análisis realizado se ha podido definir 4 módulos principales, los cuales nos ayudarán para obtener el producto deseado y requerido por el usuario:

Módulo de ingreso de horarios tutoría: en el cual se registrará el horario por materia y grupo de estudiantes en el cual el docente podrá brindar este apoyo académico a los estudiantes en riesgo de deserción.

Módulo de ingreso tutorías: el cual permitirá registrar la tutoría impartida por el docente a un grupo de estudiantes de un cierto tema específico.

Módulo de calificación tutoría: aquí el estudiante podrá calificar al docente por la tutoría recibida con la contestación de 5 preguntas que van a estar relacionadas con el desempeño, actitud y conocimiento del tutor.

Módulo de reportería: se podrá generar los reportes previamente analizados que mostrarán información acerca de:

- Asistencia de estudiantes a la tutoría
- Horarios tutoría
- Calificación de la tutoría
- Tutorías impartidas por el docente

Para el desarrollo de esta aplicación el lenguaje que se utilizará es Java (JSF) ya que posee grandes características como es: la universalidad, sencillez, orientación a objetos, seguridad extrema y diseño específico para computación en red además de poseer librerías con un costo menor de mantenimiento y mucha información de apoyo para el desarrollo de aplicaciones.

Para la creación de los reportes se utilizará la herramienta JasperSoft ya que éste me permite crear reportes de una manera muy fácil y dinámica.

El servidor web utilizado para el desarrollo de esta aplicación es GlassFish ya que permite la centralización a través de una consola de administración y de una interfaz de línea de comandos más fácil y amigable con los desarrolladores.

2.4.1 Cronograma entregables

“Los cuadros de entregables, constituyen un acuerdo entre el cliente, el cual estable la prioridad de cada historia de usuario, de acuerdo con el valor que aporta para el negocio.”
(Letelier & Sánchez, 2003)

En el cronograma de entregables están las fechas y versiones que se han realizado con las historias de usuario y tarjetas CRC.

Tabla 12. *Cronograma de entregables de historias de usuario y tarjetas CRC*

ITERACIONES	Nº	HISTORIAS DE USUARIO Y TARJETAS CRC	PRIORIDAD (ENTREGA)	ACTIVIDAD (NUEVA/CORRECCIÓN MEJORA)	VERSIÓN	ESTADO DESARROLLO	FECHA ENTREGA	ESTADO APROBACIÓN
ENTREGA DE HISTORIAS DE USUARIO	1	GESTIÓN DE SEGURIDAD	1	NUEVA	1.0	COMPLETO	21/01/2015	APROBADO
	2	GESTIÓN DE NUEVO USUARIO	1	NUEVA	1.0	COMPLETO	22/01/2015	APROBADO
	3	GESTIÓN DE PERFILES	1	NUEVA	1.0	COMPLETO	23/01/2015	APROBADO
	4	GESTIÓN DE MÓDULOS	1	NUEVA	1.0	COMPLETO	24/01/2015	APROBADO
	5	GESTIÓN DE INGRESO ESTUDIANTES	1	NUEVA	1.1	COMPLETO	25/01/2015	APROBADO
	6	GESTIÓN DE HORARIOS TUTORIAS	1	NUEVA	1.0	COMPLETO	26/01/2015	APROBADO
	7	GESTIÓN TUTORIAS	1	NUEVA	1.0	COMPLETO	27/01/2015	APROBADO
	8	GESTIÓN CALIFICACION TUTORIAS POR PARTE DEL ESTUDIANTE	1	NUEVA	1.0	COMPLETO	28/01/2015	APROBADO
	9	GESTIÓN DE REPORTES	1	NUEVA	1.0	COMPLETO	29/01/2015	APROBADO
ENTREGA DE TARJETAS CRC	10	CRC DE SEGURIDAD	2	NUEVA	1.0	COMPLETO	30/01/2015	APROBADO
	11	CRC DE INGRESO USUARIOS	2	NUEVA	1.0	COMPLETO	31/01/2015	APROBADO
	12	CRC DE PERFILES	2	NUEVA	1.0	COMPLETO	01/02/2015	APROBADO
	13	CRC DE MÓDULOS	2	NUEVA	1.0	COMPLETO	02/02/2015	APROBADO
	14	CRC INGRESO ESTUDIANTES	2	NUEVA	1.0	COMPLETO	03/02/2015	APROBADO
	15	CRC HORARIOS TUTORIAS	2	NUEVA	1.0	COMPLETO	04/02/2015	APROBADO
	16	CRC REGISTRO TUTORIAS	2	NUEVA	1.0	COMPLETO	05/02/2015	APROBADO
	17	CRC CALIFICACION TUTORIAS POR PARTE DEL ESTUDIANTE	2	NUEVA	1.0	COMPLETO	06/02/2015	APROBADO
	18	CRC REPORTES	2	NUEVA	1.0	COMPLETO	07/02/2015	APROBADO

Nota. Cronograma de entregables de historias de usuario y tarjetas CRC

Elaborado por: Diego Zurita Granizo

2.4.2 Tarjetas CRC

La técnica CRC propone una forma de trabajo, preferentemente grupal, para encontrar los objetos del dominio de la aplicación, sus responsabilidades y como colaboran con otros para realizar tareas. Esta técnica utiliza las llamadas tarjetas CRC, las cuales registran el nombre de las clases, sus responsabilidades y las otras clases con la que colaboran. Además de la forma de registrar la información de las clases, la técnica propone un proceso llamado proceso CRC.

Las principales características de las tarjetas son:

- Identificación de clases y asociaciones que participan del diseño del sistema.
- Obtención de las responsabilidades que debe cumplir cada clase.
- Establecimiento de cómo una clase colabora con otras clases para cumplir con sus responsabilidades. (Bahamondes, 2011)

Tabla 13. *Tarjeta CRC – Seguridad del Sistema*

SEGURIDAD	
Validación de datos ingresados	UsuarioSessionBean MD5Encryptor
Verificación permisos de acceso al sistema	
Devolución de datos	

Nota. Seguridad del Sistema

Elaborado por: Diego Zurita Granizo

Tabla 14. *Tarjeta CRC – Ingreso Usuario*

INGRESO USUARIOS	
Registro de Datos	UsuarioSessionBean MD5Encryptor

Nota. Ingreso Usuario

Elaborado por: Diego Zurita Granizo

Tabla 15. Tarjeta CRC - Perfiles

PERFILES	
Verificación de acceso al sistema Asignación de perfil Habilitación y des habilitación de perfiles	UsuarioSessionBean

Nota. Tarjeta CRC - Perfiles

Elaborado por: Diego Zurita Granizo

Tabla 16. Tarjeta CRC – Módulos

MÓDULOS	
Validación de datos ingresados Devolución de datos Verificación de permisos de acceso al sistema	UsuarioSessionBean

Nota. Tarjeta CRC – Módulos

Elaborado por: Diego Zurita Granizo

Tabla 17. Tarjeta CRC – Ingreso de Estudiantes

INGRESO ESTUDIANTES	
Verificación de acceso al sistema	UsuarioSessionBean
Ingreso de datos	EstudiantesSessionBean
Actualización de información	MD5Encryptor
Eliminación de la información	EstudiantesSessionBean

Nota. Tarjeta CRC – Ingreso de Estudiantes

Elaborado por: Diego Zurita Granizo

Tabla 18. *Tarjeta CRC – Horarios Tutorías*

HORARIOS TUTORIAS	
Verificación de acceso al sistema	UsuarioSessionBean
Ingreso de datos	ProfesoresSessionBean
Actualización de información	MD5Encryptor
Eliminación de la información	ProfesoresSessionBean

Nota. Tarjeta CRC – Horarios Tutorías

Elaborado por: Diego Zurita Granizo

Tabla 19. *Tarjeta CRC – Registro de la Tutorías*

REGISTRO TUTORIAS	
Verificación de acceso al sistema	UsuarioSessionBean
Ingreso de datos	ProfesoresSessionBean
Actualización de información	MD5Encryptor
Eliminación de la información	ProfesoresSessionBean

Nota. Tarjeta CRC – Registro de la Tutorías

Elaborado por: Diego Zurita Granizo

Tabla 20. *Tarjeta CRC – Calificación de la Tutoría por parte del Estudiantes*

CALIFICACION TUTORIA POR PARTE DEL ESTUDIANTE	
Verificación de acceso al sistema	UsuarioSessionBean
Ingreso de datos	ProfesoresSessionBean
Actualización de información	MD5Encryptor
Eliminación de la información	ProfesoresSessionBean

Nota. *Tarjeta CRC – Calificación de la Tutoría por parte del Estudiantes*

Elaborado por: Diego Zurita Granizo

Tabla 21. *Tarjeta CRC - Reportes*

REPORTES	
Verificación de permisos de acceso al sistema	ReporteControlador
Devolución de Datos	

Nota. Tarjeta CRC - Reportes

Elaborado por: Diego Zurita Granizo

2.4.3 Unified Modeling Language (UML)

“Es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido concebido por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh.” (Ferré & Sánchez, 2011, pág. 1)

“El UML es una de las herramientas más emocionantes en el mundo actual del desarrollo de sistemas. Esto se debe a que permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicárselas a otras personas” (Schmuller, 2001, pág. 3)

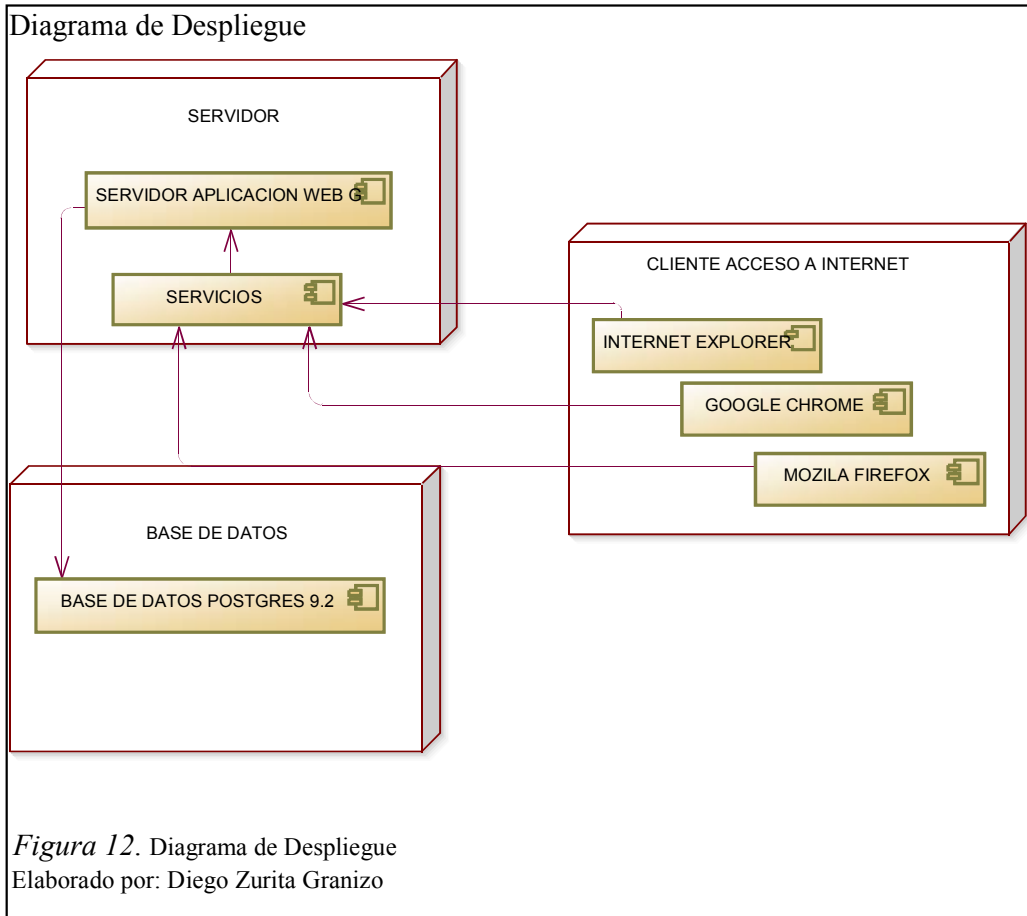
El UML no solo se basa en diagramas de casos de uso y diagramas de clases los cuales ya se mencionaron, para un mayor entendimiento de la arquitectura de la aplicación se realizó el diagrama de despliegue.

2.4.3.1 Diagrama de despliegue:

Este diagrama muestra la arquitectura física de un sistema informático. Se intenta representar equipos y dispositivos, interconexiones y el software que se encontrará en cada máquina. (Schmuller, 2001, pág. 94)

Los elementos que lo componen son:

- **Nodo:** elemento físico que suele contar con memoria y capacidad de procesamiento.
- **Artefacto:** elemento que puede ser ejecutado dentro de un nodo.
- **Conectores:** representan conexiones físicas entre nodos (esto incluye las inalámbricas).
- **Dispositivo:** nodo sin capacidad de procesamiento.
- **Ambiente de Ejecución (Executive Environment):** un nodo que ofrece un ambiente de ejecución. (Zuluaga, 2008)



2.5 Diseño de la Base de Datos

La herramienta de diseño para el modelamiento de la base de datos se llama PowerDesigner la cual permite de una manera más fácil analizar y manipular los datos y así lograr una efectiva arquitectura de información, el motor de base de datos utilizado para el desarrollo de este proyecto es PostgreSQL ya que este permite un bases de datos relacionales orientados a objetos y además es de libre distribución y es un requerimiento del usuario la utilización de este motor de base de datos.

[illegible]

Elaborado por: Diego Zurita Granizo

2.5.2 Modelo Físico

Modelo físico de la base de datos

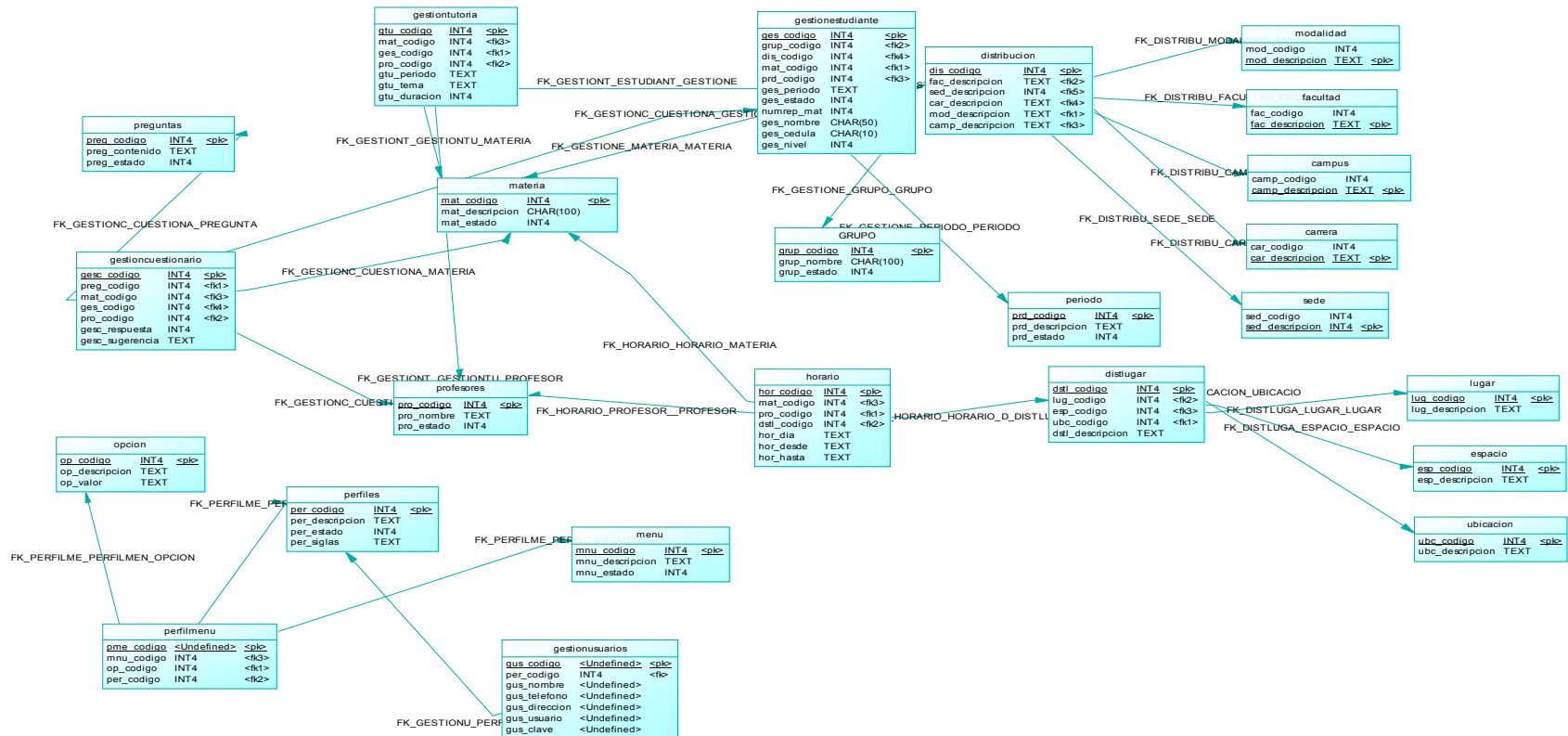


Figura 14. Modelo Físico de la base de datos

Elaborado por: Diego Zurita Granizo

2.5.3 Diccionario de datos

En el diccionario de datos se describirán todas las tablas de la base de datos llamada “gestionaca” con sus respectivos campos con su tipo de dato y la descripción correspondiente a cada uno de estos.

Tabla 22. *Diccionario de datos de la tabla campus*

Tabla: campus		Fecha de Creación: 25-01-2015
Descripción: Contiene la información de los campus de la Universidad Politécnica Salesiana.		
Campo	Tipo	Descripción
camp_codigo	Integer	Clave secuencial
camp_descripcion	Character	Descripción
Claves camp_codigo		Primaria:

Nota. Tabla de ingreso de los campus

Elaborado por: Diego Zurita Granizo

Tabla 23. *Diccionario de datos de la tabla modalidad*

Tabla: modalidad		Fecha de Creación: 25-01-2015
Descripción: Contiene la información de las modalidades de matrícula.		
Campo	Tipo	Descripción
mod_codigo	Integer	Clave secuencial
mod_descripcion	Character	Descripción
Claves Primaria: mod_codigo		

Nota. Tabla de ingreso de los campus

Elaborado por: Diego Zurita Granizo

Tabla 24. *Diccionario de datos de la tabla facultad*

Tabla: facultad		Fecha de Creación: 25-01-2015
Descripción: Contiene la información de las áreas que existen en la Universidad Politécnica Salesiana.		
Campo	Tipo	Descripción
fac_codigo	Integer	Clave secuencial
fac_descripcion	Character	Descripción
Claves Primaria: fac_codigo		

Nota. Tabla de ingreso de los campus

Elaborado por: Diego Zurita Granizo

Tabla 25. *Diccionario de datos de la tabla carrera*

Tabla: carrera		Fecha de Creación: 25-01-2015
Descripción: Contiene la información de las carreras que existen en la Universidad Politécnica Salesiana.		
Campo	Tipo	Descripción
car_codigo	Integer	Clave secuencial
car_descripcion	Character	Descripción
Claves Primaria: car_codigo		

Nota. Tabla de ingreso de las carreras

Elaborado por: Diego Zurita Granizo

Tabla 26. Diccionario de datos de la tabla sede

Tabla: carrera		Fecha de Creación: 25-01-2015
Descripción: Contiene la información de las sedes que existen en la Universidad Politécnica Salesiana.		
Campo	Tipo	Descripción
sed_codigo	Integer	Clave secuencial
sed_descripcion	Character	Descripción
Claves Primaria: sed_codigo		

Nota. Tabla de ingreso de sede

Elaborado por: Diego Zurita Granizo

Tabla 27. Diccionario de datos de la tabla distribución

Tabla: tb_profesor		Fecha de Creación: 25-01-2015
Descripción: Contiene la información personal de cada docente.		
Campo	Tipo	Descripción
dis_codigo	Integer	Clave secuencial
fac_codigo	Integer	Clave de la facultad
sed_codigo	Integer	Clave de la sede
car_codigo	Integer	Clave de la carrera
mod_codigo	Integer	Clave de la modalidad
camp_codigo	Integer	Clave del campus
Claves Foráneas: facultad con el campo fac_cod ,dis_codigo Sede con el campo sed_cod Carrera con el campo car_cod Modalidad con el campo mod_cod Capus con el campo camp_cod		Clave Primaria:

Nota. Tabla de ingreso de la distribución

Elaborado por: Diego Zurita Granizo

Tabla 28. *Diccionario de datos de la tabla lugar*

Tabla: lugar		Fecha de Creación: 25-01-2015
Descripción: Contiene la información de los lugares donde realizar una tutoría.		
Campo	Tipo	Descripción
lug_codigo	Integer	Clave secuencial
lug_descripcion	Character	Descripción
Claves Primaria: Lug_codigo		

Nota. Tabla de ingreso de los lugares

Elaborado por: Diego Zurita Granizo

Tabla 29. *Diccionario de datos de la tabla espacio*

Tabla: espacio		Fecha de Creación: 25-01-2015	Descripción:
Contiene la información de las aulas donde se realizarán las tutoría.			
Campo	Tipo	Descripción	
esp_codigo	Integer	Clave secuencial	
esp_descripcion	Character	Descripción	
Claves Primaria: esp_codigo			

Nota. Tabla de ingreso de los espacios

Elaborado por: Diego Zurita Granizo

Tabla 30. *Diccionario de datos de la tabla ubicación*

Tabla: ubicación		Fecha de Creación: 25-01-2015	Descripción:
Contiene la información de ubicación donde se realizarán las tutoría.			
Campo	Tipo	Descripción	
esp_codigo	Integer	Clave secuencial	
esp_descripcion	Character	Descripción	
Claves Primaria:			
esp_codigo			

Nota. Tabla de ingreso de la ubicación

Elaborado por: Diego Zurita Granizo

Tabla 31. *Diccionario de datos de la tabla distlugar*

Tabla: distlugar		Fecha de Creación: 25-01-2015
Descripción: Contiene la información de la distribución de los lugares para la tutoría.		
Campo	Tipo	Descripción
dstl_codigo	Integer	Clave secuencial
lug_codigo	Integer	Clave de la facultad
esp_codigo	Integer	Clave de la sede
ubc_codigo	Integer	Clave de la carrera
dstl_descripcion	Integer	Clave de la modalidad
Claves Foráneas:		Clave Primaria:
facultad con el campo fac_cod		
dis_codigo		
Sede con el campo sed_cod		
Carrera con el campo car_cod		
Modalidad con el campo mod_cod		
Capus con el campo camp_cod		

Nota. Tabla de ingreso de las distribuciones del lugar

Elaborado por: Diego Zurita Granizo

Tabla 32. *Diccionario de datos de la tabla materia*

Tabla: materia		Fecha de Creación: 25-01-2015	Descripción:
Contiene la información de las materias que van a realizar tutoría los docentes.			
Campo	Tipo	Descripción	
mat_codigo	Integer	Clave secuencial	
mat_descripcion	Character	Descripción	
mat_estado	Integer	Estado del registro 1: activo; 2:eliminado	
Claves Primaria:			
mat_codigo			

Nota. Tabla de ingreso de las materias

Elaborado por: Diego Zurita Granizo

Tabla 33. *Diccionario de datos de la tabla grupo*

Tabla: grupo		Fecha de Creación: 25-01-2015
Descripción: Contiene la información de los grupos a los que pertenece un estudiante		
Campo	Tipo	Descripción
grup_codigo	Integer	Clave secuencial
grup_nombre	Character	Descripción
grup_estado	Integer	Estado del registro 1: activo; 2:eliminado
Claves Primaria: grup_codigo		

Nota. Tabla de ingreso de los grupos

Elaborado por: Diego Zurita Granizo

Tabla 34. *Diccionario de datos de la tabla gestionestudiante*

Tabla: gestionestudiante		Fecha de Creación: 25-01-2015	Descripción:
Contiene la información de los estudiantes que van a recibir la tutoría.			
Campo	Tipo	Descripción	
ges_codigo	Integer	Clave secuencial	
grup_codigo	Integer	Clave del grupo	
dis_codigo	Integer	Clave de la distribución	
mat_codigo	Integer	Clave de la materia	
prd_codigo	Integer	Clave del período	
ges_estado	Integer	Estado del registro 1: activo; 2:eliminado	
numrep_mat	Integer	Número de veces que repite un estudiantes la materia	
ges_nombre	Character	Nombre del estudiantes	
ges_cedula	Character	Cédula del estudiante	
ges_nivel	Integer	Nivel del estudiante	
Claves Foráneas: grupo con el campo grup_cod ges_codigo distribución con el campo dis_cod materia con el campo mat_cod período con el campo prd_cod			Clave Primaria:

Nota. Tabla de ingreso de los estudiantes

Elaborado por: Diego Zurita Granizo

Tabla 35. *Diccionario de datos de la tabla gestiontutoria*

Tabla: gestiontutoria		Fecha de Creación: 25-01-2015
Descripción: Contiene la información de las tutoría impartidas.		
Campo	Tipo	Descripción
gtu_codigo	Integer	Clave secuencial
mat_codigo	Integer	Clave de la materia
ges_codigo	Integer	Clave del estudiante
pro_codigo	Integer	Clave de la profesor
prd_codigo	Integer	Clave del período
gtu_tema	Integer	Tema tratado en la tutoría
gtu_duracion	Integer	Duración de la tutoría períodos de 15 minutos.
Claves Foráneas:		Clave Primaria:
estudiante con el campo ges_cod		gtu_codigo
profesor con el campo pro_cod		
materia con el campo mat_cod		
período con el campo prd_cod		

Nota. Tabla de ingreso de la tutoría
Elaborado por: Diego Zurita Granizo

Tabla 36. *Diccionario de datos de la tabla profesores*

Tabla: profesores		Fecha de Creación: 25-01-2015
Descripción: Contiene la información de los profesores que impartirán la tutoría.		
Campo	Tipo	Descripción
pro_codigo	Integer	Clave secuencial
pro_nombre	Integer	Nombre de profesor
pro_estado	Integer	Clave del estudiante
Claves Primaria:		
pro_codigo		

Nota. Tabla de ingreso de los profesores
Elaborado por: Diego Zurita Granizo

Tabla 37. *Diccionario de datos de la tabla horario*

Tabla: profesores		Fecha de Creación: 25-01-2015	Descripción:
Contiene la información de los profesores que impartirán la tutoría.			
Campo	Tipo	Descripción	
hor_codigo	Integer	Clave secuencial	
mat_codigo	Integer	Clave de la materia	
pro_codigo	Integer	Clave del profesor	
dstl_codigo	Integer	Estado del registro 1: activo; 2:eliminado	
hor_dia	Character	El día que realizo la tutoría	
hor_desde	Character	La hora desde	
hor_hasta	Character	La hora hasta	
Claves Foráneas:		Clave Primaria:	
profesor con el campo pro_cod		hor_codigo	
materia con el campo mat_cod			
distribución con el campo dstl_cod			

Nota. Tabla de ingreso del horario
Elaborado por: Diego Zurita Granizo

Tabla 38. *Diccionario de datos de la tabla pregunta*

Tabla: pregunta		Fecha de Creación: 25-01-2015	Descripción: Contiene la información de las preguntas que se muestran para la calificación de la tutoría.
Campo	Tipo	Descripción	
preg_codigo	Integer	Clave secuencial	
preg_contenido	Integer	Pregunta	
preg_estado	Integer	Estado del registro 1: activo; 2:eliminado	
Claves Principal: preg_codigo			

Nota. Tabla de ingreso de las preguntas
Elaborado por: Diego Zurita Granizo

Tabla 39. *Diccionario de datos de la tabla gestioncuestionario*

Tabla: gestioncuestionario Fecha de Creación: 25-01-2015 Descripción: Contiene la información de la calificación realizada por los estudiantes a la tutoría recibida.		
Campo	Tipo	Descripción
gesc_codigo	Integer	Clave secuencial
preg_codigo	Integer	Clave de la pregunta
mat_codigo	Integer	Clave del materia
ges_codigo	Integer	Clave del estudiante
pro_codigo	Integer	Clave del profesor
gesc_respuesta	Character	Respuesta
gesc_sugerencia	Character	Sugerencia
Claves Foráneas: Clave Primaria: profesor con el campo pro_cod gesc_codigo materia con el campo mat_cod preguntas con el campo preg_cod gestionestudiante con el campo ges_cod		

Nota. Tabla de ingreso de la contestación de las preguntas

Elaborado por: Diego Zurita Granizo

Tabla 40. *Diccionario de datos de la tabla opción*

Tabla: opcion Fecha de Creación: 25-01-2015		
Descripción: Contiene la información de las preguntas que se muestran para la calificación de la tutoría.		
Campo	Tipo	Descripción
op_codigo	Integer	Clave secuencial
op_descripcion	Character varying	Descripción de la opción
op_valor	Integer	Estado del registro 1: activo; 2:eliminado
Claves Principal: op_codigo		

Nota. Tabla de ingreso de las opciones del menú

Elaborado por: Diego Zurita Granizo

Tabla 41. *Diccionario de datos de la tabla perfiles*

Tabla: perfiles		Fecha de Creación: 25-01-2015
Descripción: Contiene los perfiles que maneja la aplicación		
Campo	Tipo	Descripción
per_codigo	Integer	Clave secuencial
per_descripcion	Character varying	Descripción del perfil
per_estado	Integer	Estado del registro 1: activo; 2:eliminado
per_siglas	Character varying	Descripción de las siglas del perfil
Claves Principal: per_codigo		

Nota. Tabla de ingreso de los perfiles

Elaborado por: Diego Zurita Granizo

Tabla 42. *Diccionario de datos de la tabla perfilmenu*

Tabla: perfilmenu		Fecha de Creación: 25-01-2015	Descripción:
Contiene las opciones del menú que le corresponde a cada perfil			
Campo	Tipo	Descripción	
pem_codigo	Integer	Clave secuencial	
mnu_codigo	Character varying	Clave del menú	
op_codigo	Integer	Clave de la opción	
per_codigo	Character varying	Clave del perfil	
Claves Foráneas:		Clave Primaria:	
menu con el campo mnu_codigo		pem_codigo	
opcion con el campo op_codigo			
perfiles con el campo per_cod			

Nota. Tabla de ingreso de los perfiles al menú

Elaborado por: Diego Zurita Granizo

Tabla 43. Diccionario de datos de la tabla menú

Tabla: perfilmenu		Fecha de Creación: 25-01-2015
Descripción: Contiene las opciones del menú que le corresponde a cada perfil		
Campo	Tipo	Descripción
mnu_codigo	Integer	Clave secuencial
mnu_descripcion	Character varying	Descripción
mnu_estado	Integer	Estado del registro 1: activo; 2:eliminado
Claves Principal: mnu_codigo		

Nota. Tabla de ingreso de menús

Elaborado por: Diego Zurita Granizo

Tabla 44. Diccionario de datos de la tabla gestionusuarios

Tabla: gestionusuarios		Fecha de Creación: 25-01-2015	Descripción:
Contiene la información de los usuarios.			
Campo	Tipo	Descripción	
gus_codigo	Integer	Clave secuencial	
per_codigo	Integer	Descripción	
gus_nombre	Character	Nombre del usuario	
gus_telefono	Character	Teléfono	
gus_direccion	Character	Dirección	
gus_usuario	Character	Usuario	
gus_clave	Character	contraseña	
Claves Foráneas:		Clave Primaria:	
perfiles con el campo per_cod		gus_codigo	

Nota. Tabla de ingreso de la gestión de los usuarios

Elaborado por: Diego Zurita Granizo

2.6 Análisis de factibilidad

Después de definir la problemática presente y establecer las causas que ameritan de un nuevo sistema que permita el registro y control de las tutorías académicas, es necesario realizar un estudio de factibilidad para determinar la infraestructura tecnológica y la capacidad técnica que implica la implementación del sistema en cuestión así como los costos, beneficios y el grado de aceptación que la propuesta genera en la Universidad Politécnica Salesiana, los aspectos tomados en cuenta para este estudio fueron clasificados en 2 áreas, las cuales se describen a continuación:

2.6.1 Factibilidad técnica:

La factibilidad técnica nos permitió realizar una recolección de información sobre los componentes técnicos existentes en la Universidad Politécnica Salesiana y la posibilidad de hacer uso de los mismos en la implementación del sistema propuesto, para lo cual se ha evaluado 2 enfoques: Hardware y Software.

2.6.1.1 Hardware

Los requisitos mínimos que deben cumplir el servidor donde va a ser instalada la aplicación son los siguientes:

- Adaptadores de red.
- Procesador de 1.66GHz o superior.
- Memoria mínima de 1Gb.

Evaluando el hardware existente y tomando en cuenta los requisitos mínimos necesarios, la Universidad Politécnica Salesiana no necesitará realizar ninguna inversión para la adquisición de equipos nuevos y tampoco para repotenciar o actualizar equipos existentes.

El servidor disponible en la Universidad Politécnica Salesiana posee las siguientes características:

- Memoria: 3.7 GB
- Procesador: Intel (R) Xeon (R) X5650 @ 2.6 GHz
- Espacio disponible en disco: 81,2 GB

2.6.1.2 Software

En cuanto al Software la Universidad Politécnica Salesiana cuenta con su servidor con el sistema operativo Linux. Para el uso general del aplicativo se debe poseer los navegadores que existen actualmente en el mercado.

Tabla 45. *Software disponible*

Cantidad	Descripción
01	Sistema Operativo Linux, Versión Centos 6.5
01	PostgreSql 7.1
01	Servidor de aplicaciones GlassFish 3.1.2

Nota. Software disponible

Elaborado por: Diego Zurita Granizo

2.6.2 Factibilidad económica:

En la factibilidad económica podemos mencionar que el nuevo aplicativo a desarrollarse es muy rentable desde el punto de vista de la obtención de información de una manera más ágil y confiable.

La Universidad no tendrá que realizar ningún gasto adicional para la implementación de este aplicativo ya que cuenta con el hardware y software necesarios. Los costos de desarrollo no están involucrados ya que es un proyecto de tesis.

A continuación se muestra un aproximado del valor ahorrado por la Universidad en el desarrollo de este aplicativo referente al pago de un desarrollador senior.

Tabla 46. *Costo de desarrollo senior*

Costo-hora	Tiempo de desarrollo	# de horas de desarrollo	Costo Total
\$20	3 meses	2196	\$43.900

Nota. Costos del desarrollador senior

Elaborado por: Diego Zurita Granizo

2.6.2.1 Análisis costos-beneficios

Este análisis permite realizar una comparación entre la relación de los costos que genera el proceso actual de tutorías académicas y los costos que tendría con la implementación del nuevo aplicativo, conociendo los beneficios que la ciencia de la informática ofrece. Como se mencionó anteriormente en el estudio de factibilidad técnica la universidad ya cuenta con las herramientas necesarias para trabajar con el aplicativo propuesto en esta tesis, por lo que la implementación de este aplicativo no necesitara de una inversión inicial.

Costos Generales

Los gastos generales se encuentran representados o enmarcados por todos aquellos gastos en accesorios y materiales de oficina de uso diario, necesarios para realizar los procesos, tales como: esferos, papel, cuadernos, cartuchos de impresora.

Tabla 47. Costos de papelería y material de oficina

Gastos generales	Costo aproximado USD	Consumo mensual	Monto anual en USD
Material de oficina	3	2	72
Papel para impresión	2	2	48
Cartuchos de impresión	25	1	300
Total	30	6	420

Nota. Costos de papelería y materia de oficina

Elaborado por: Diego Zurita Granizo

Costos del aplicativo web propuesto

El aplicativo permitirá el registro y control de las tutorías académicas donde involucra los mismos costos generales descritos en la tabla 46, pero al lograr optimizar los procesos y agilitando el flujo y manejo de la información de las actividades de seguimiento y control de las tutorías académicas la Dirección de Carrera podrá tomar acciones inmediatas a problemas que se susciten con los estudiantes en dicho apoyo académico, ya así lograr un índice menor de pérdidas de materia o deserciones en la carrera.

CAPÍTULO 3

IMPLEMENTACIÓN Y PRUEBAS

3.1 Requerimientos de ejecución

Se conoce como requerimientos de ejecución aquellos que permiten que las interfaces de la aplicación trabajen de una manera más eficaz, eficiente e independientemente.

3.1.1 Interfaces de hardware

Es necesario que los equipos cumplan con las siguientes características:

- Adaptadores de red.
- Procesador de 1.66GHz o superior.
- Memoria mínima de 1Gb.

3.1.2 Interfaces de software

- **Sistema Operativo:** Windows XP o superior.
- **Explorador:** Internet Explorer, Mozilla Firefox o Chrome.

3.1.3 Interfaces de comunicación

El servidor que se utilizará para cargar la aplicación tiene instalado el sistema operativo Centos 6.5 con núcleo Linux 2.6, donde se encuentran instalados los siguientes programas para su correcto funcionamiento:

- Base de datos PostgreSQL 9.2
- Servidor Web Glassfish 3.1.2

3.1.4 Interfaces de hardware del servidor

El servidor que se utilizará posee las siguientes características:

- Memoria: 3.7 GB
- Procesador: Intel (R) Xeon (R) X5650 @ 2.6 GHz
- Espacio disponible en disco: 81,2 GB

3.2 Seguridad

En la encriptación de la información se utilizó el algoritmo MD5 el cual se basa en la codificación de 128 bits el cual se identifica como un número de 32 dígitos hexadecimales.

Para la encriptación de cadenas este algoritmo utiliza cuatro pasos importantes que son:

1. Adición de bits
2. Longitud del mensaje
3. Inicializar el búfer MD
4. Procesado del mensaje en bloques de 16 palabras

Código Java – Código Encriptación MD5

```
public class Utils {  
    public static String encriptar(String message, String algorithm){  
        byte[] digest = null;  
        byte[] buffer = message.getBytes();  
        try {  
            MessageDigest messageDigest = MessageDigest.getInstance(algorithm);  
            messageDigest.reset();  
            messageDigest.update(buffer);  
            digest = messageDigest.digest();  
        } catch (NoSuchAlgorithmException ex) {  
            System.out.println("Error creando Digest");  
        }  
        return toHexadecimal(digest);  
    }  
    private static String toHexadecimal(byte[] digest){  
        String hash = "";  
        for(byte aux : digest) {  
            int b = aux & 0xff;  
            if (Integer.toHexString(b).length() == 1) hash += "0";  
            hash += Integer.toHexString(b);  
        }  
        return hash;  
    }  
}
```

If-Else Statements Must Use Braces
.....
(Alt-Enter shows hints)

Figura 15. Código Java – Código Encriptación MD5

Elaborado por: Diego Zurita Granizo

Código Java – Implementación encriptación en la aplicación web

```

objetoProfesor = profesoresService.validaProfesor(registroEmp[11].getContents().toString());
if (objetoProfesor == null) {
    objetoProfesor = new Profesores();
    objetoProfesor.setProNombre(registroEmp[11].getContents().toString());
    objetoProfesor.setProEstado(1);
    profesoresService.agregaProfesor(objetoProfesor);
    //ingreso el usuario en la tabla
    int cont=0;
    for(String resp:registroEmp[11].getContents().toString().split(" ")){
        if(cont==0){
            GestionUsuarios user=null;
            user=usuariosService.validaUsuario(registroEmp[11].getContents().toString(), registroEmp[11].getContents().toString());
            if(user==null){
                Perfiles perfil=new Perfiles();
                perfil=usuariosService.buscarPerfil("doc");
                user=new GestionUsuarios();
                user.setGusCodigo(null);
                user.setGusUsuario(resp);
                user.setGusClave(Utils.encryptar(resp,"MD5"));
                user.setPerCodigo(perfil);
                user.setGusNombre(registroEmp[11].getContents().toString());
                usuariosService.agregarUsuario(user);
            }
            break;
        }
        cont++;
    }
}

```

Figura 16. Código Java – Implementación encriptación en la aplicación web
Elaborado por: Diego Zurita Granizo

Encriptación de claves – Base de Datos

gus_codigo serial	gus_nombre character(10)	gus_telefono character(20)	gus_direccion character(10)	gus_usuario character(20)	gus_clave text	per_codigo integer
3815	RAMIREZ MON			RAMIREZ	E7040c59032bdf215419c2bdd8b13c1e	3
3816	VARGAS JACC			VARGAS	d5c75824afd9a8768486d41fcc5f0f01	3
3817	VANEGAS TOA			VANEGAS	97e3ebaae71ee09f30a242574f4041c4	3
3819	NAVAS RUILO			NAVAS	8de2b00947f5a8d2a259f567dfd70fe7	3
3821	LOPEZ NAVAR			LOPEZ	c4841d890dedae0088dae925d8e1ae70	3
3822	TIJILIMA CI			TIJILIMA	2b0cea27cad995b60362051146be23e6	3
3823	GAIBOR NARA			GAIBOR	64d9f91c506628802d56b6443f816f6a	3
3824	TIJILIMA CI			TIJILIMA	7791ab00b8e9117f9ab7ac5a7fd9a5911	3

Figura 17. Encriptación de claves – Base de Datos
Elaborado por: Diego Zurita Granizo

3.3 Mantenimiento

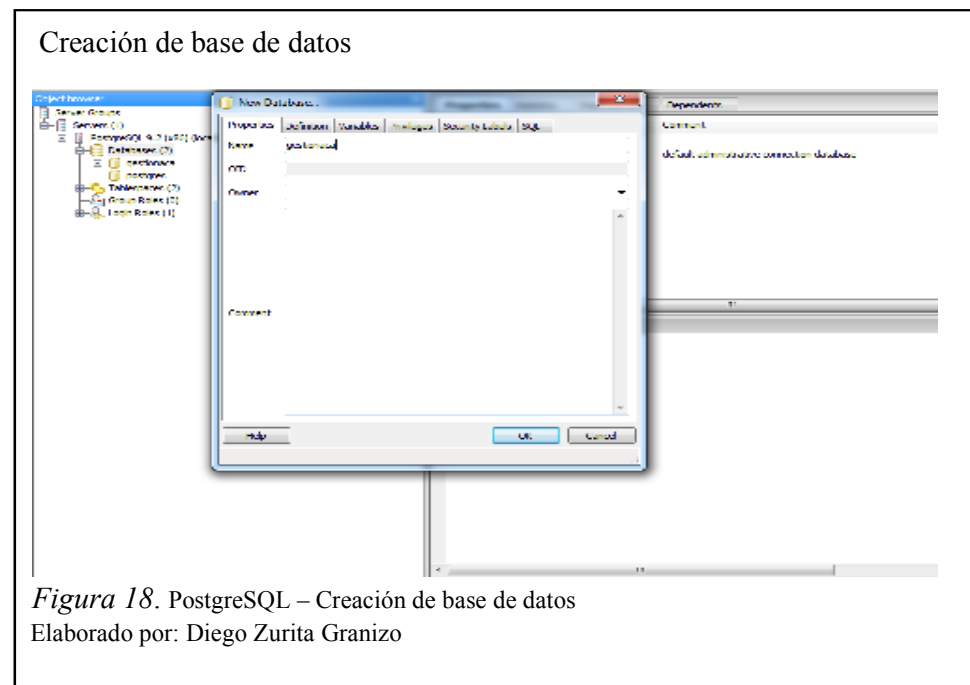
Para realizar el mantenimiento del aplicativo será necesario tener el perfil de administrador para ello se realizará el manual de instalación y el de usuario del aplicativo para su optima utilización.

3.3.1 Manual de instalación

Es necesario tener instalado PostgreSQL y Glassfish en el servidor que se vaya a subir el aplicativo en desarrollo.

Base de datos en PostgreSQL

Dentro de PostgreSQL se crea una nueva base de datos con el nombre gestionaca.



Después de la creación la base de datos se procede con la restauración del backup con lo que se restauran todas las tablas y la información necesaria que necesita el sistema.

[illegible]

Elaborado por: Diego Zurita

Base de datos (gestionaca)

The screenshot shows the Visual Studio IDE with a C# project named 'Project1'. The Solution Explorer on the left lists the project files, including 'Program.cs'. The main editor area displays the code for 'Program.cs', which is a simple console application that prints 'Hello, World!'. The Properties window on the right shows the 'AssemblyName' property set to 'Project1'.

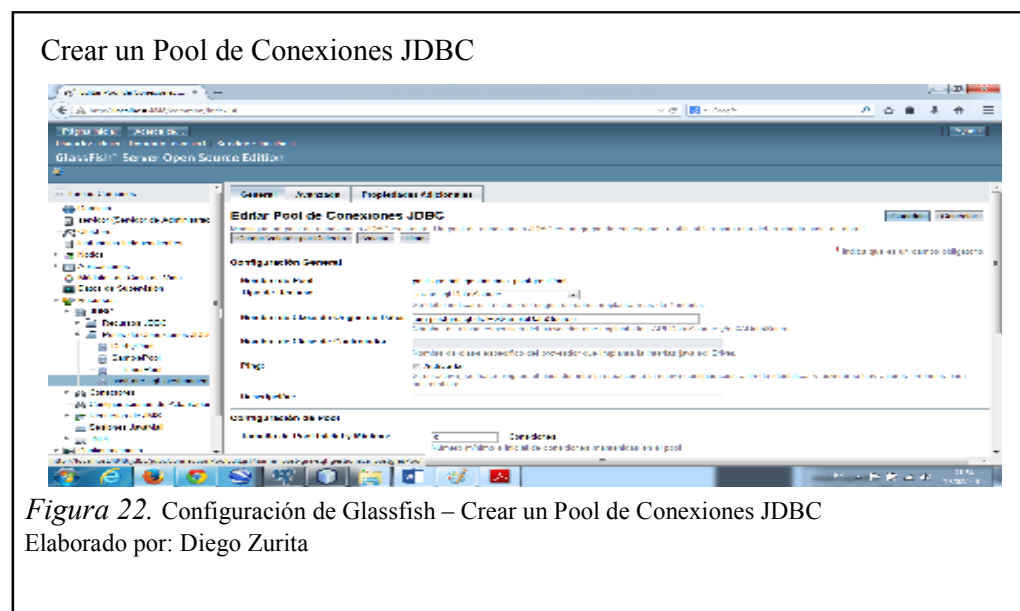
Elaborado por: Diego Zurita Granizo

Configuración de Glassfish

Para acceder a la interfaz gráfica de Glassfish y proceder con la configuración del mismo, en el navegador web se coloca la siguiente dirección: <http://localhost:4848>



Para realizar la conexión con la base de datos y la aplicación se debe crear primero un Pool de Conexiones JDBC (Java Database Connectivity), el cual se definirá con el nombre de “post-gre-sql_gestionaca_postgresPool”



Dentro de JDBC existe la opción Recursos JDBC donde se debe crear un nuevo Recurso llamado jdbc_gestion.

Configuración de Glassfish – Recursos JDBC

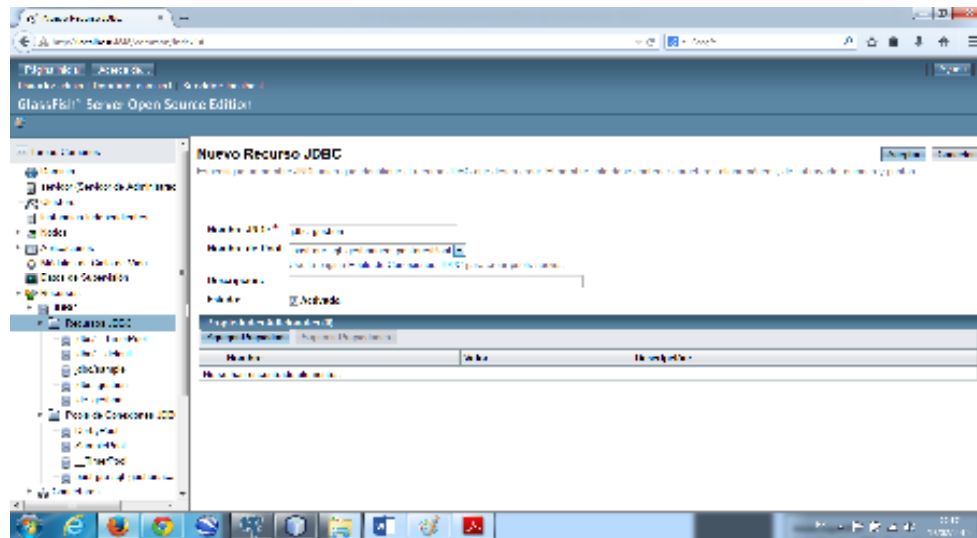


Figura 23. Configuración de Glassfish – Recursos JDBC

Elaborado por: Diego Zurita

Configuración de Glassfish – Verificación del Recurso JDBC creado

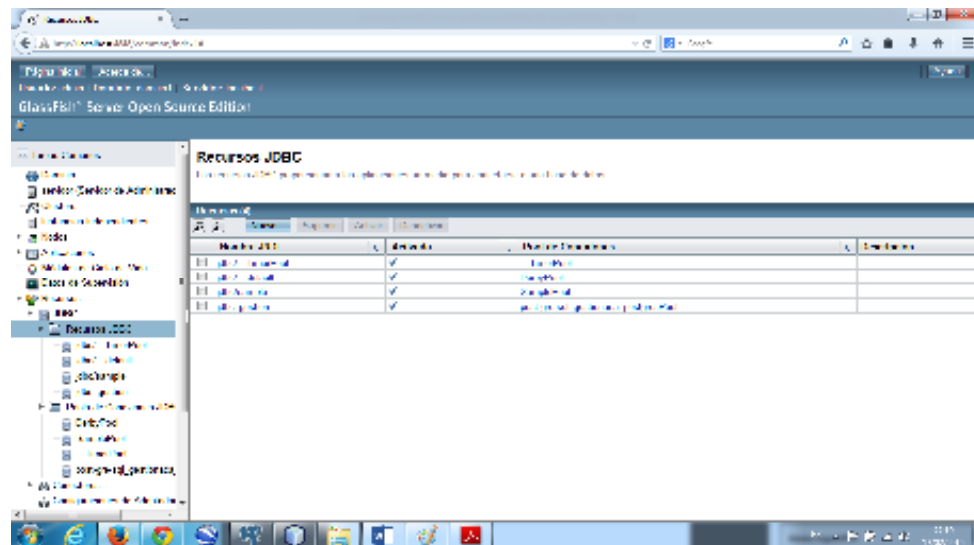


Figura 24. Configuración de Glassfish – Verificación del Recurso JDBC creado

Elaborado por: Diego Zurita

Hibernate

Dentro del Glassfish en la opción Herramienta de Actualización se debe buscar Hibernate e instalarla ya que esta herramienta nos ayudara con el Mapeo objeto-relacional de nuestra base de datos.

Instalación de Hibernate

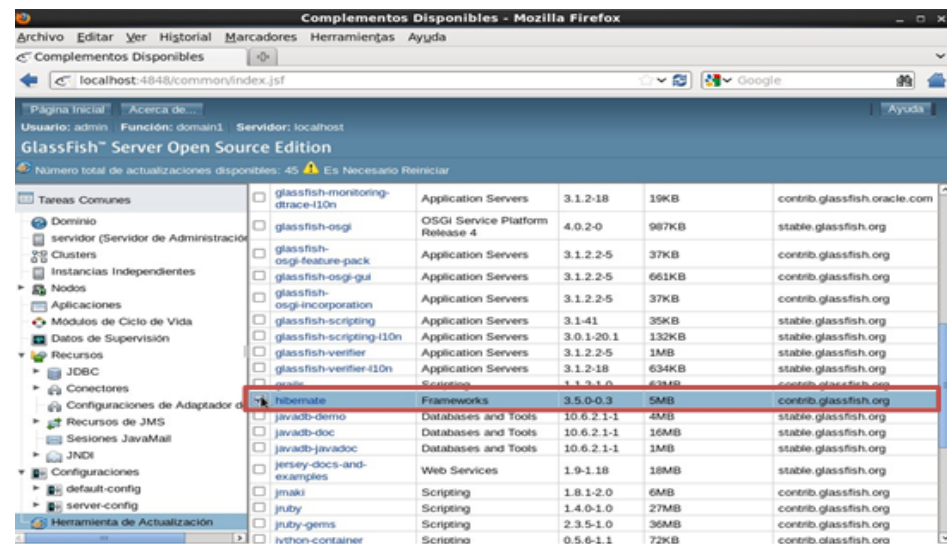


Figura 25. Instalación de Hibernate

Elaborado por: Diego Zurita

Luego de instalar Hibernate debe reiniciar el Glassfish.

Instalación de Hibernate

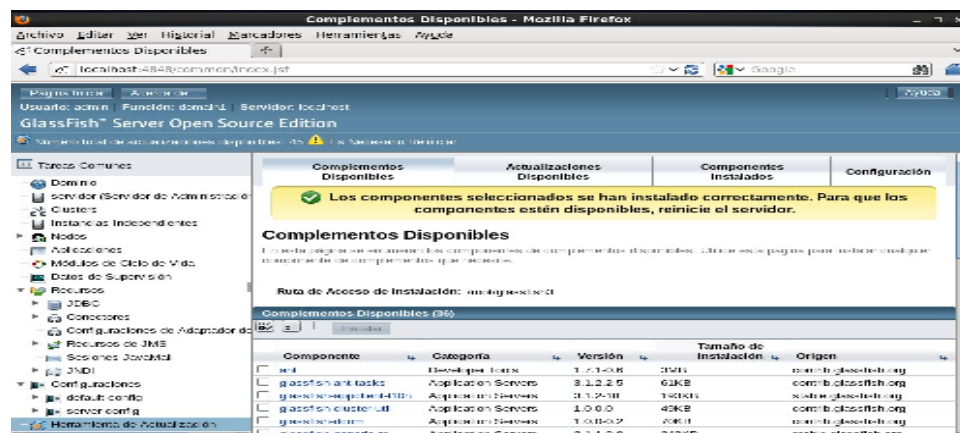


Figura 26. Instalación de Hibernate – Mensaje de confirmación

Elaborado por: Diego Zurita

Archivos WAR y JAR

Estos archivos son generados desde Eclipse y para que se despliegue el ambiente web es necesario copiar ambos archivos a la ruta:

`\root\glassfish3\glassfish\domains\domain1\autodeploy`

Archivos WAR y JAR en la ruta especificada



Figura 27. Archivos WAR y JAR en la ruta especificada

Elaborado por: Diego Zurita

Después de copiar ambos archivos, en la interfaz de administración de Glassfish nos ingresamos a la opción Aplicaciones, y le indicamos el archivo que se desea deployar.

Deployar el archivo JAR

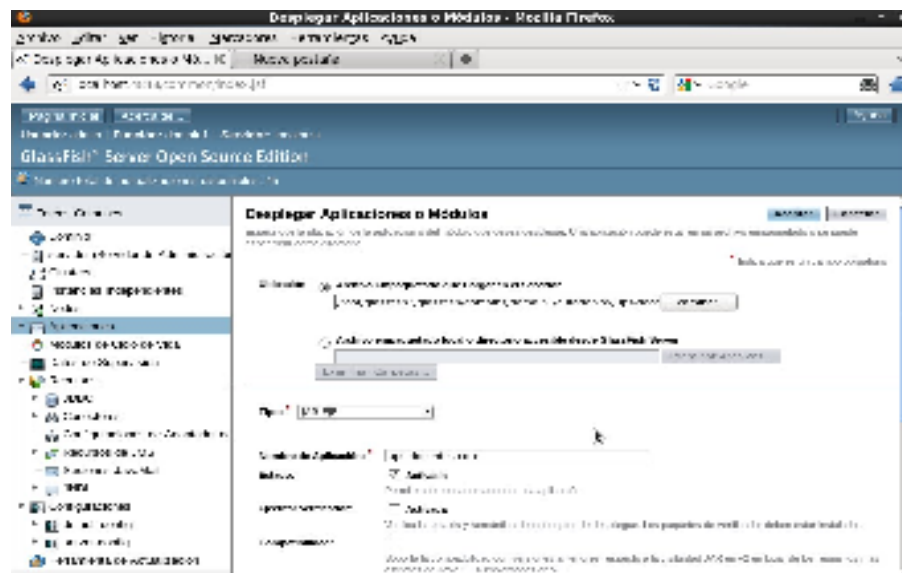


Figura 28. Archivos JAR – Deployar el archivo JAR

Elaborado por: Diego Zurita

Figura 29. Archivos WAR – Deployar el archivo WAR

Elaborado por: Diego Zurita

UNIVERSIDAD POLITÉCNICA SALESIANA
INNOVACIÓN

ENTREVISTA A TUTORÍAS ACADÉMICAS

ALUMNO

FECHA

ENTREVISTA ENTREVISTA NUEVA

Tutorías Académicas

Se trata de un servicio que ofrece a los estudiantes la oportunidad de recibir orientación académica y profesional de parte de los docentes de la institución.

3.3.2 Manual de usuario

Para poder ingresar al aplicativo de Tutorías Académicas debemos autenticarnos, para esto nos solicitará que ingresemos nuestro usuario y contraseña.

Autenticación

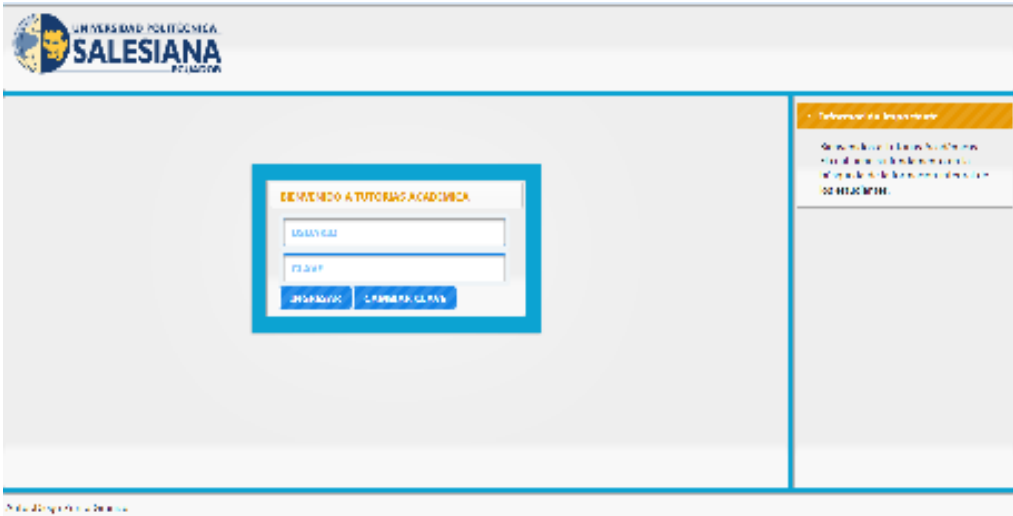


Figura 31. Autenticación
Elaborado por: Diego Zurita

El campo usuario: debemos ingresar el usuario en el caso de un perfil estudiante es su número de cédula para un perfil docente su usuario es su apellido paterno en mayúsculas y para el administrador es adm.

El campo Clave: debemos ingresar la clave que nos haya sido asignado, en el caso de ser estudiante la clave es su número de cédula, si es un docente la clave es su apellido paterno en mayúsculas y para administrador es adm.

El botón ingresar: verifica que los datos en los campos usuario y clave correspondan algún registro dentro de la base de datos, si no lo encontrara mostrará un mensaje informando que el usuario no es válido.

EL botón cambiar clave: nos permite realizar el cambio de clave asignado, con el ingreso de los campos solicitados.

Luego de autenticarnos dependiendo de nuestro perfil ingresaremos a los siguientes módulos:

Módulo de Docentes: este módulo tiene 3 submódulos de ingreso que son:

- Horarios
- Cuestionario
- Tutoría

Los cuales nos sirven para ingresar información necesaria con la que se podrá generar los reportes que este módulo tiene habilitado y que son los siguientes:

- Reporte de Horarios
- Reporte Evaluaciones
- Reporte Tutorías

Submódulo 1.- Horarios: aquí nos permite realizar el ingreso de los horarios del docente en los cuales va a impartir la tutoría, todos los campos son obligatorios, cuando el docente comienza a escoger los días que brindará la tutoría el sistema comienza a validar que el mínimo de horas por día para las tutorías sea de una hora.

Para realizar el ingreso del horario de las tutorías académicas el docente inicialmente escogerá la materia donde debe brindar las tutorías, luego el grupo que son los estudiantes que están relacionados a la materia antes escogida la hora y el día, y el espacio físico de la tutoría que comprende el lugar, la ubicación y el espacio. El botón “grabar” realizará el ingreso de la información a la base de datos. El botón “nuevo” reiniciara las variables que se utilizan para permitir el ingreso de un nuevo registro.

Materia: se muestra las materias asignadas a un docente en el período vigente.

Grupo: muestra los grupos asignados a la materia seleccionada.

Hora desde: hora inicial a ser seleccionada.

Hora hasta: hora final a ser seleccionada.

Día: muestra los días de la semana que se podrá ingresar un horario para la tutoría.

Lugar: muestra los lugares disponibles para brindar una tutoría.

Ubicación: carga las ubicaciones disponibles con el lugar previamente seleccionado.

Espacio: carga los espacios disponibles con la ubicación previamente seleccionada.

Horario

Figura 32. Horario
Elaborado por: Diego Zurita

Submódulo 2.- Cuestionario: aquí el docente podrá ingresar sus comentarios acerca de los estudiantes que han asistido a la tutoría, estos comentarios son de manera individual. En este submódulo solo aparecerán los estudiantes que hayan asistido a una tutoría como mínimo.

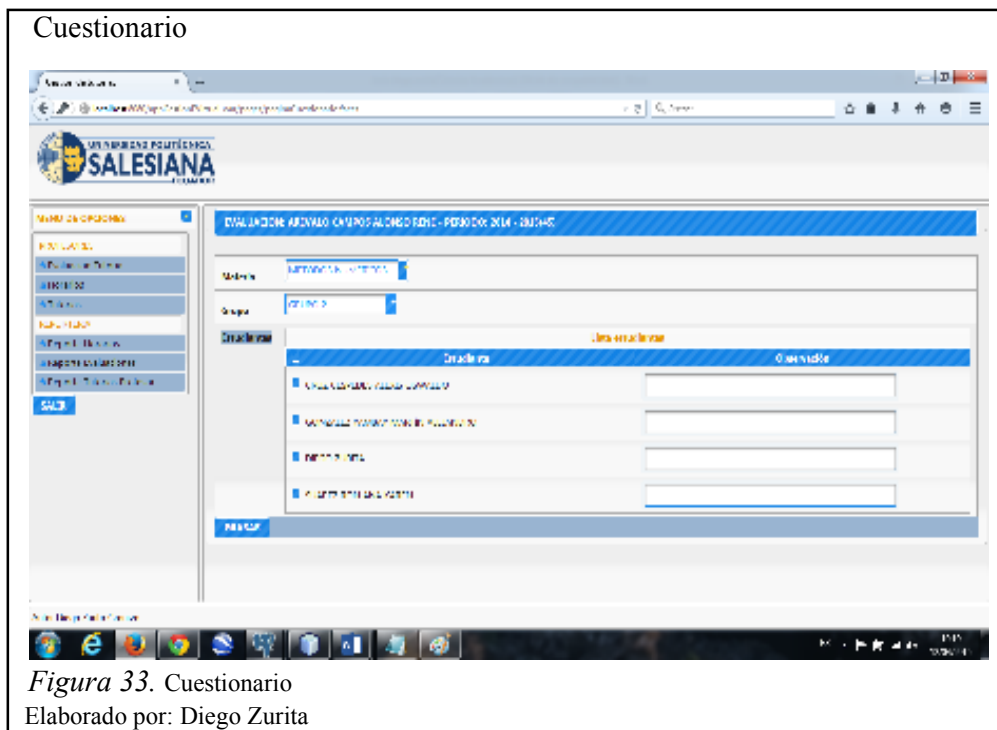
Materia: se muestra las materias asignadas a un docente en el periodo vigente.

Grupo: muestra los grupos asignados a la materia seleccionada.

Campo de observación: ingreso de la observación del docente con respecto de un estudiante que ha tomado una tutoría.

Lista estudiantes: muestra los estudiantes que pertenecen a la materia y grupo previamente seleccionados que hayan recibido una tutoría.

Botón grabar: ingresará la información solicitada en la base de datos.



Submódulo 3.- Tutoría: se podrá ingresar la tutoría por estudiante o grupo de estudiantes todos los campos son obligatorios, dentro de este submódulo se tiene la opción de “Estudiante no Asignado”, el cual permite ingresar a un estudiante que no está en riesgo de deserción y que ha tomado voluntariamente la tutoría, de igual manera se tiene la opción de ver los horarios que el docente ha ingresado previamente y un breve historial de los alumnos que se mostrara al dar click sobre el nombre de ellos.

Botón ver horarios: nos muestra los horarios del docente que ha ingresado al aplicativo.

Materia: se muestra las materias asignadas a un docente en el período vigente.

Grupo: muestra los grupos asignados a la materia seleccionada.

Lista estudiantes: muestra los estudiantes que pertenecen a la materia y grupo previamente seleccionados.

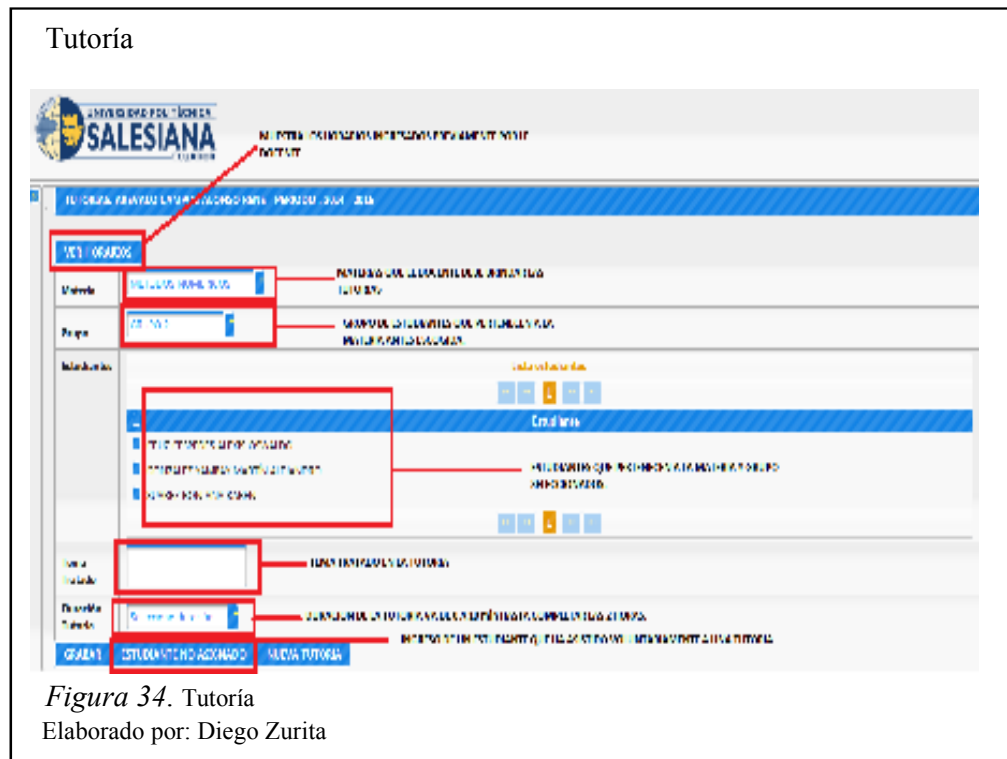
Tema tratado: ingreso del tema que se trató en la tutoría.

Duración tutoría: tiempo que se tardó en brindar una tutoría.

Botón grabar: ingresa la información seleccionada a la base de datos.

Botón estudiante no asignado: ingresa a un estudiante que haya recibido voluntariamente la tutoría.

Botón nueva tutoría: reinicia los valores para el ingreso de una nueva tutoría.



Horario dentro de la tutoría



Figura 35. Horario dentro de la tutoría

Elaborado por: Diego Zurita

Estudiante no asignado

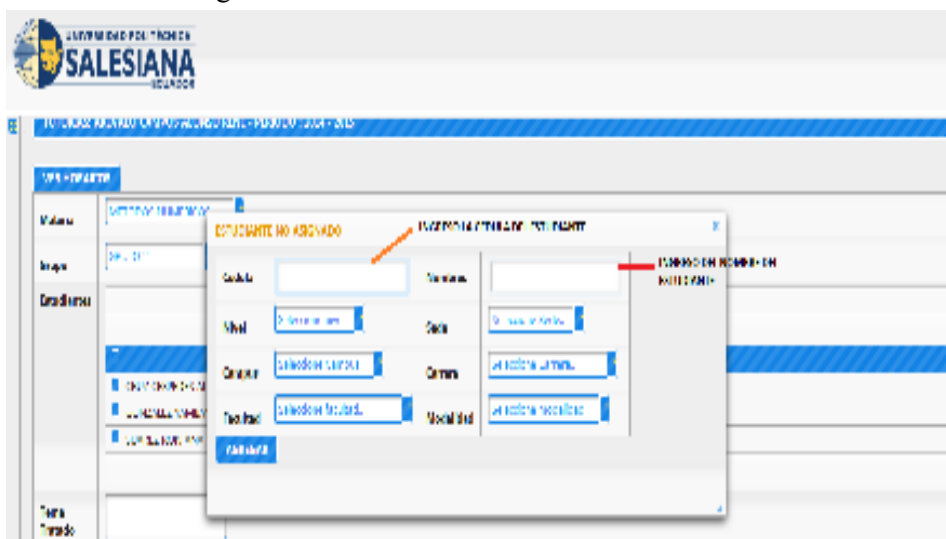


Figura 36. Estudiante no asignado

Elaborado por: Diego Zurita

Historial estudiante

Docente	Variable	Denominación	Turno
AREVALO CAMPOS ALONSO RENE	EXAMEN DE HISTORIA		

No. Registro	Nombre	Periodo
7	INSTRUMENTOS DE EVALUACION	2014 - 2015
8	UNIDAD DE EVALUACION	2014 - 2015
9	TECNOLOGIA	2014 - 2015

Figura 37. Historial estudiante

Elaborado por: Diego Zurita

Reporte horarios: este mostrará los horarios del docente que han sido ingresados en el período actual con la facilidad de poder imprimirlos.

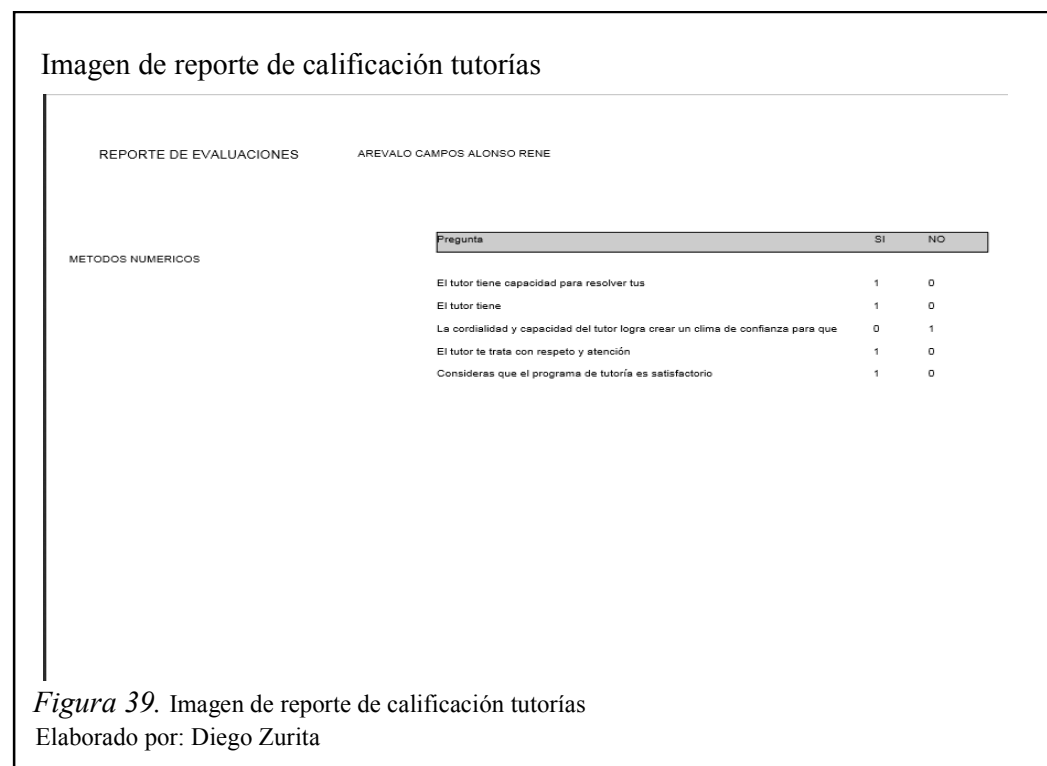
Reporte de horarios docentes

REPORTE DE HORARIOS DE TODOS LOS PROFESORES						
MATERIA	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO
AREVALO CAMPOS ALONSO RENE						
METODOS NUMERICOS	13:30 - 14:00 LABORATORIO BLOQUE A AULA 1	13:30 - 14:00 LABORATORIO BLOQUE A AULA 1	13:30 - 14:00 LABORATORIO BLOQUE A AULA 1			
SISTEMAS GERENCIALES (G)	13:30 - 14:00 LABORATORIO BLOQUE A AULA 1					

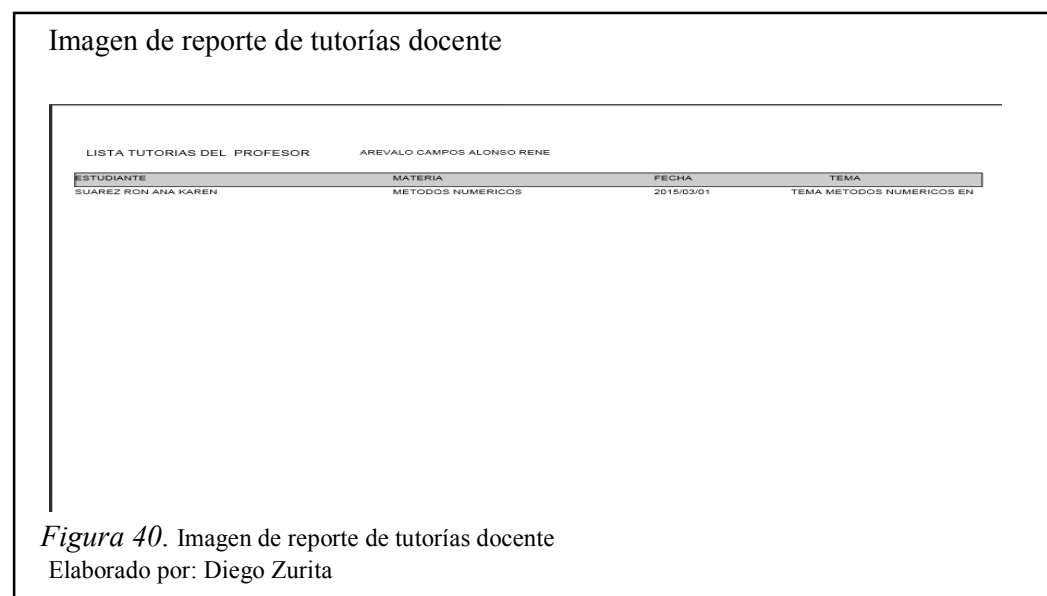
Figura 38. Reporte de horarios docentes

Elaborado por: Diego Zurita

Reporte evaluaciones: este reporte mostrará las evaluaciones que ha tenido un docente al impartir una tutoría.



Reporte Tutorías Profesor: este mostrará todas las tutorías que un docente ha realizado en un período determinado.



- Cuestionario: en el cual el estudiante podrá calificar la tutoría que recibió por parte de un docente, escogerá el profesor y la materia y contestará las preguntas que están relacionadas al desempeño y conocimiento del docente.

- Reporte de Tutorías Estudiantes: el cual mostrará la información de cuantas tutorías asistió el alumno en el período actual, y que permita ser impreso.

[illegible]

Reporte de tutorías estudiante



Figura 42. Reporte de tutorías estudiante
Elaborado por: Diego Zurita

Módulo de Administrador: este módulo contiene 8 submódulos de configuración del sistema que son:

- Perfiles
- Encuestas
- Carga archivo
- Períodos
- Asignación tutoría
- Carrera-Facultades
- Ubicación-Espacio
- Menú

Y los reportes que puede generar son

- Reporte de Horarios
- Reporte Evaluaciones
- Reporte Tutorías
- Reporte Asistencia Estudiante

Cuando se ingresa a este módulo la página de inicio nos muestra varios reportes gráficos que contienen información como:

- La relación en número de los estudiantes contra los docentes en ese período actual.
- Las materias con segunda matrícula.
- Las materias con tercera matrícula.
- Los profesores que ya han brindado tutorías.

Módulo administrador

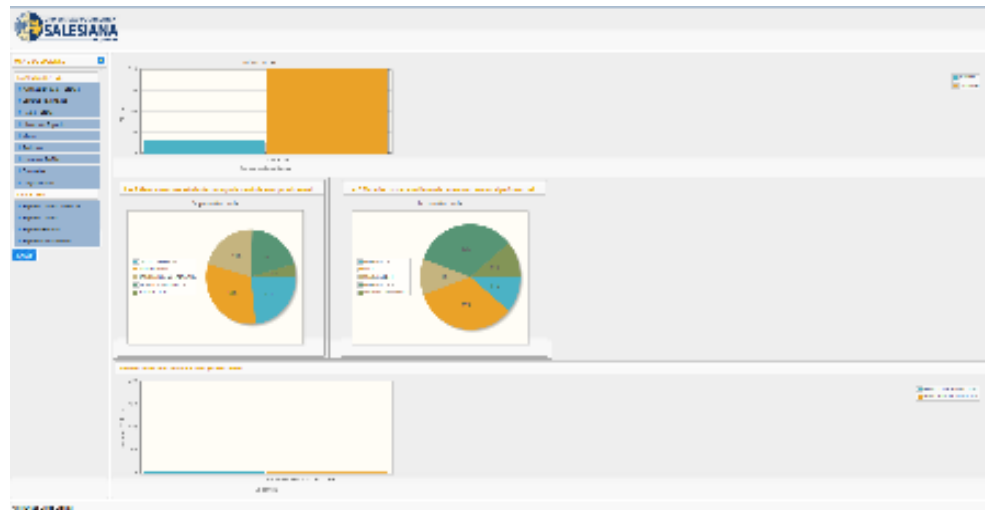


Figura 43. Página de inicio módulo administrador

Elaborado por: Diego Zurita

Submódulo 1.- Perfiles: dentro de este módulo podemos encontrar a los usuarios con su información principal las acciones que nos permite realizar son el actualizar información, encontramos también los perfiles que maneja el aplicativo los cuales podemos actualizar y eliminarlos.

Nombres: se debe ingresar el nombre del nuevo usuario.

Teléfono: se debe ingresar el teléfono.

Dirección: se debe ingresar la dirección de domicilio o trabajo.

Perfil: se debe seleccionar el perfil con el que el nuevo usuario trabajará en el aplicativo.

Usuario: se debe ingresar el usuario con el que se autenticará en el aplicativo, solo se permite números y letras, máximo de 20 caracteres.

Clave: se debe ingresar la clave del nuevo usuario, solo se permite números y letras, máximo de 32 caracteres.

Figura 44. Usuarios

Contenido pregunta: ingreso de la pregunta, solo se permiten caracteres, máximo 200.

Botón Grabar: graba la información solicitada en el formulario.

Al hacer click en la pregunta aparecerá una ventana en donde nos permite activar o deshabilitar a la misma, con la opción “estado pregunta”.

Encuestas

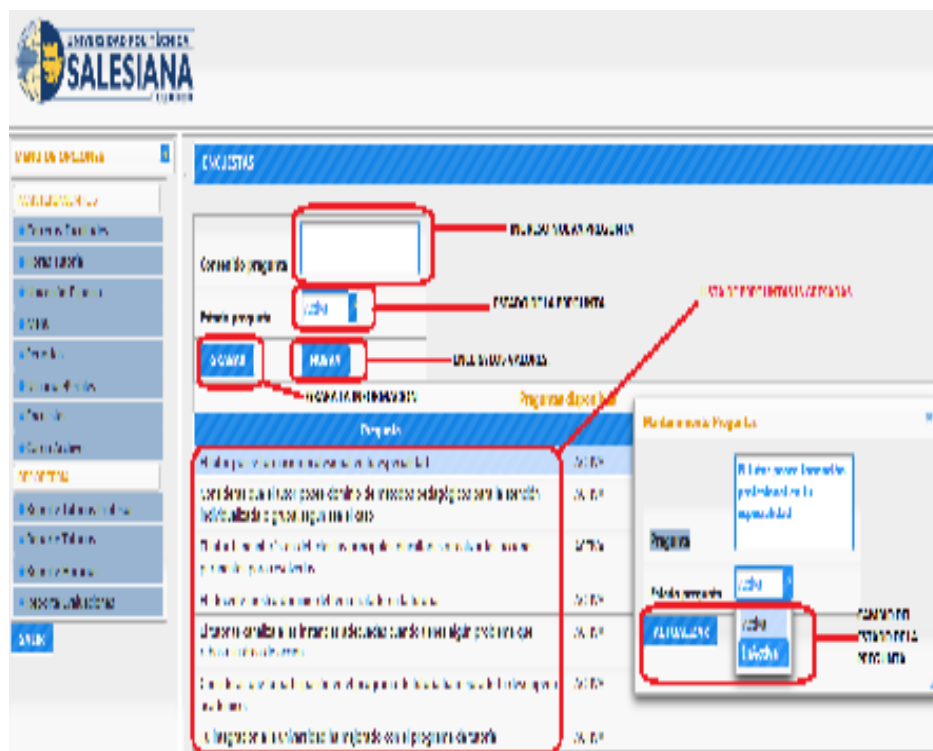


Figura 45. Página encuestas

Elaborado por: Diego Zurita

Submódulo 3.- Carga Archivo: dentro de este módulo podemos realizar la subida del archivo en Excel que es enviado a semestralmente a la Dirección de Carrera con los estudiantes en riesgo de deserción y que con el cual todo el aplicativo comienza a funcionar.

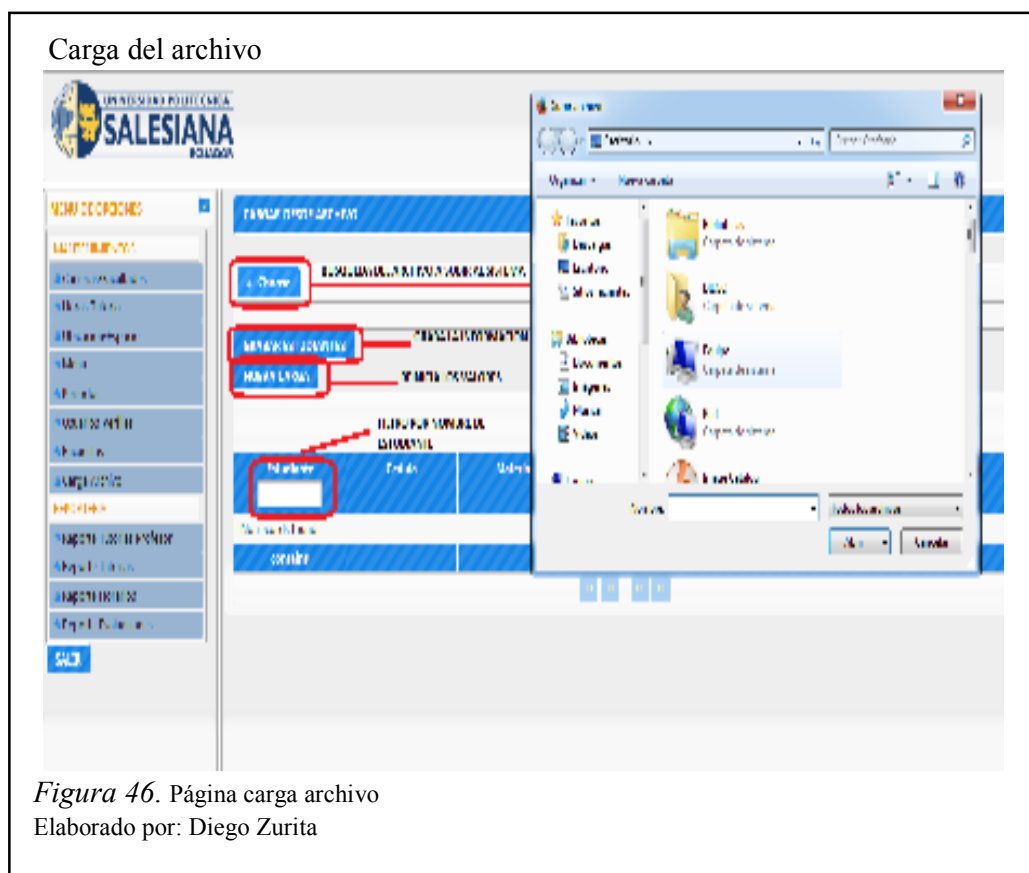
Botón choose: permite la búsqueda del archivo de Excel que contiene la información de los estudiantes de segunda y tercera matrícula.

Botón grabar estudiantes: permite grabar la información de los estudiantes que hayan pasado todas las validaciones realizadas al momento de cargar la información.

Validaciones:

- El archivo de Excel debe tener extensión .xls
- Que exista un período activo, la descripción del período debe ser igual a la que se encuentra en el archivo de Excel.
- Que el estudiante no se encuentre ingresado en el período vigente.

- Que exista la distribución de: sede, campus, carrera, área y modalidad ingresada en el aplicativo.



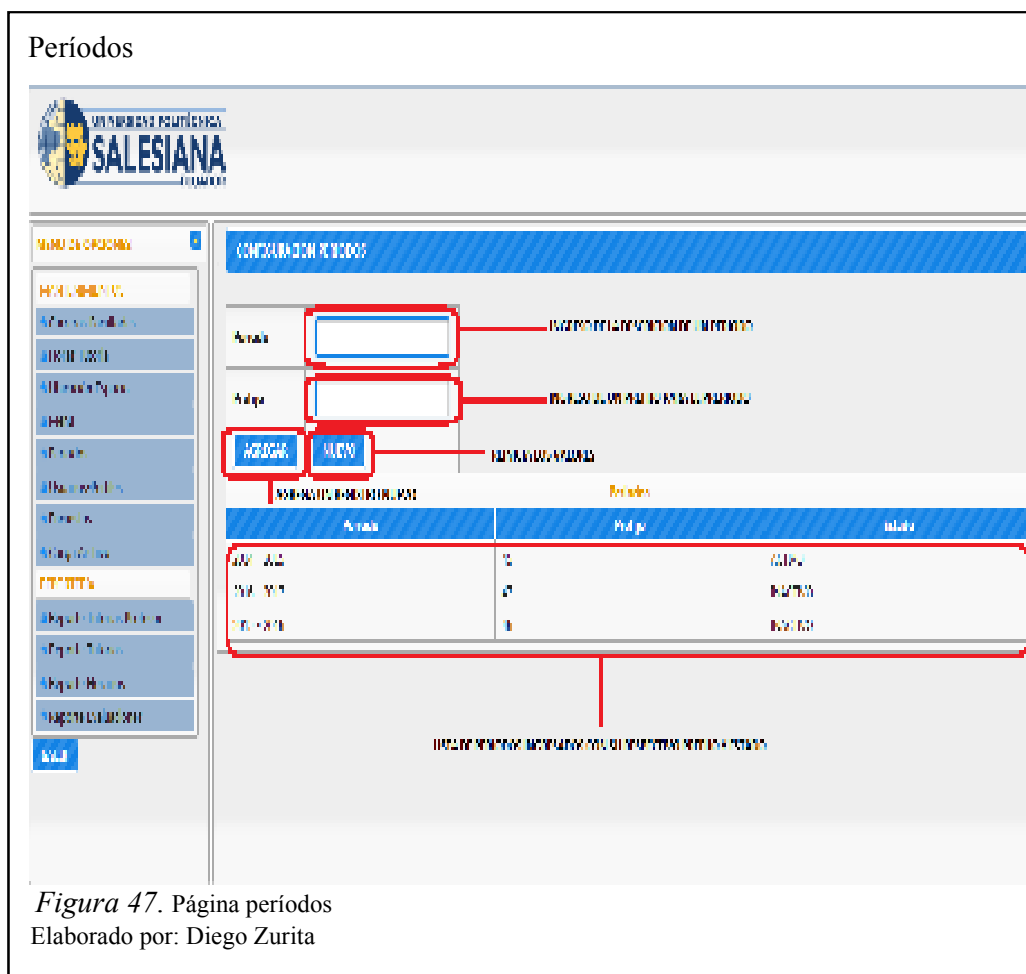
Submódulo 4.- Períodos: dentro de este módulo podemos realizar el mantenimiento de los períodos, cuando se coloca un período como predeterminado automáticamente el anterior se coloca como inactivo, se tiene también la opción de ingresar un prefijo.

Período: ingreso de la descripción de un período.

Prefijo: ingreso de un prefijo con el que se conocerá al período ingresado.

Botón agregar: guarda la información ingresada en el formulario.

Botón nuevo: reinicia los valores para el ingreso de un nuevo período.



Submódulo 5.- Carrera-Facultades: dentro de este módulo podemos realizar el mantenimiento de las sedes, campus, áreas, carreras y modalidades que necesitamos. Esta configuración es crucial al momento de subir el archivo en Excel a nuestro sistema ya que irá validando si existe la distribución que intenta subir en nuestra base de datos.

Sede: selección de las sedes ingresadas en el aplicativo.

Campus: selección de los campus que pertenezcan a la sede antes seleccionada.

Carrera: selección de las carreras que pertenezcan al campus antes seleccionado.

Área: selección de las áreas de desarrollo que pertenezcan a las carreras antes seleccionadas

Modalidad: selección de la modalidad, presencial ó semipresencial.

Botón grabar: guarda la información ingresada en el formulario.

Botón nuevo: reinicia los valores para el ingreso de una nueva distribución.

Botón actualizar: actualiza la información de la distribución seleccionada.

Botón agregar/modificar: agrega o modifica la información de los registros seleccionados.

Distributivo de carreras

Figura 48. Página de carrera-facultad
Elaborado por: Diego Zurita

Submódulo 6.- Ubicación-Espacio: dentro de este módulo podemos realizar el mantenimiento del lugar en donde se va a realizar la tutoría.

Lugar: selección del lugar para brindar la tutoría.

Ubicación: selección de la ubicación que pertenezcan al lugar previamente escogido.

Espacio: selección del espacio que pertenezcan a la ubicación previamente escogido.

Botón grabar: guarda la información ingresada en el formulario.

Botón nuevo: reinicia los valores para el ingreso de un nuevo registro.

Botón actualizar: actualiza la información de la ubicación seleccionada.

Botón agregar/modificar: agrega o modifica la información de los registros seleccionados.

Distributivo del lugar de la tutoría

Lugar	Materia	Grupo
UNIVERSIDAD	UNIVERSIDAD	UNIVERSIDAD
UNIVERSIDAD	UNIVERSIDAD	UNIVERSIDAD
UNIVERSIDAD	UNIVERSIDAD	UNIVERSIDAD

Figura 49. Página de ubicación-espacio
Elaborado por: Diego Zurita

Submódulo 7.- Asignación para tutorías: dentro de este módulo podemos realizar la asignación de los profesores que vayan a brindar las tutorías académicas de las diferentes materias que se han ingresado desde el archivo de Excel antes mencionado.

Profesor: selección de los docentes que se encuentran en el archivo en Excel y que han sido ingresados.

Grupo: selección de los grupos correspondientes a los estudiantes.

Materia: selección de las materias que se encuentran en el archivo en Excel.

Botón grabar: guarda la información ingresada en el formulario.

Botón reporte: genera un reporte de las asignaciones de tutorías realizadas.

Asignación tutoría



Figura 50. Página de asignación tutoría
Elaborado por: Diego Zurita

Reporte

UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR		
PERIODO	MATERIA	GRUPO
PROFESOR	ANÁLISIS CAMBIO ALONDO RENE	
	MÉTODOS NUMÉRICOS	GRUPO 1
	ANÁLISIS CAMBIO ALONDO RENE	GRUPO 1
PROFESOR	PADILLAS VASCO RIGOBERTO	
	CÁLCULO DIFERENCIAL	GRUPO 2
	CÁLCULO DIFERENCIAL	GRUPO 1
	CÁLCULO DIFERENCIAL	GRUPO 1

Figura 51. Reporte de la asignación tutoría
Elaborado por: Diego Zurita

Submódulo 8.- Menú: dentro de este módulo podemos realizar el mantenimiento del menú de acceso para los diferentes perfiles.

Perfiles: selección del tipo de perfil que se encuentra ingresado en el sistema.

Cabecera menú: selección de las cabeceras del menú de acceso.

Submenú: selección del submenú correspondiente a la cabecera y perfil previamente seleccionados.

Botón modificar: actualiza la información previamente seleccionada.

Al dar click en el registro aparecerá una pantalla que nos permite cambiar el estado del ítem seleccionado.

Asignación de menú



Figura 52. Página de la asignación menú
Elaborado por: Diego Zurita

Reporte Horarios: este mostrará los horarios ingresados por los docentes con la facilidad de poder imprimirlos, y podrá ser filtrado por profesor y período. El botón “Profesores” mostrará la lista de docentes para generar el reporte.

Período: selección de los períodos ingresados en el aplicativo sin importar su estado.

Profesores: selección de los profesores que han sido asignados para brindar la tutoría académica.

Botón imprimir: genera el reporte con los filtros seleccionados.

Reporte horarios

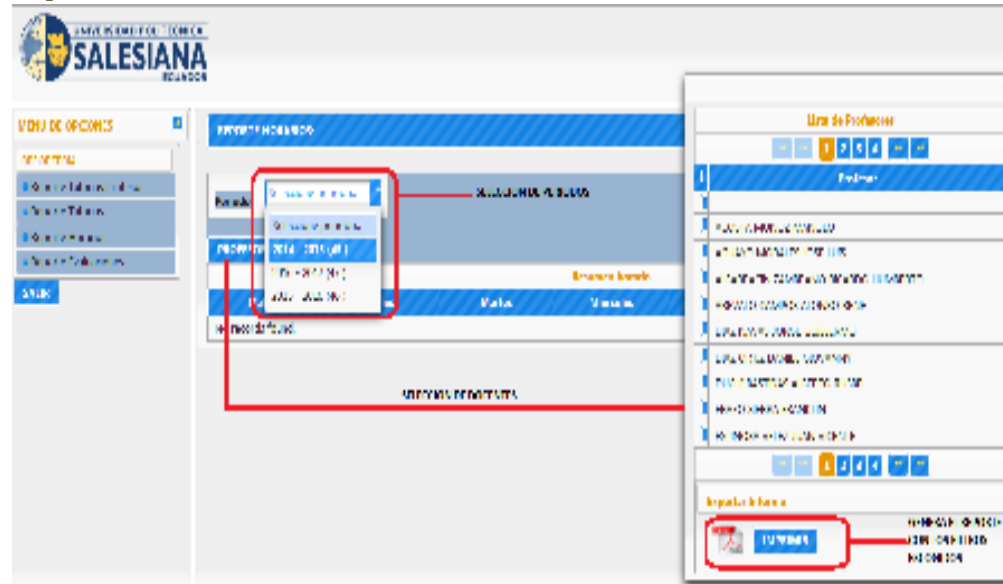


Figura 53. Imagen de filtro de horarios
Elaborado por: Diego Zurita

Reporte

REPORTE DE HORARIOS DE TODOS LOS PROFESORES						
MATERIA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SABADO
AREVALO CAMPOS ALONSO RENE						
MÉTODOS NUMÉRICOS	13:30 - 14:00 LABORATORIO BLOQUE A AULA 1	13:30 - 14:00 LABORATORIO BLOQUE A AULA 1	13:30 - 14:00 LABORATORIO BLOQUE A AULA 1	13:30 - 14:00 LABORATORIO		
SISTEMAS GERENCIALES (G)	13:30 - 14:00 LABORATORIO BLOQUE A AULA 1					

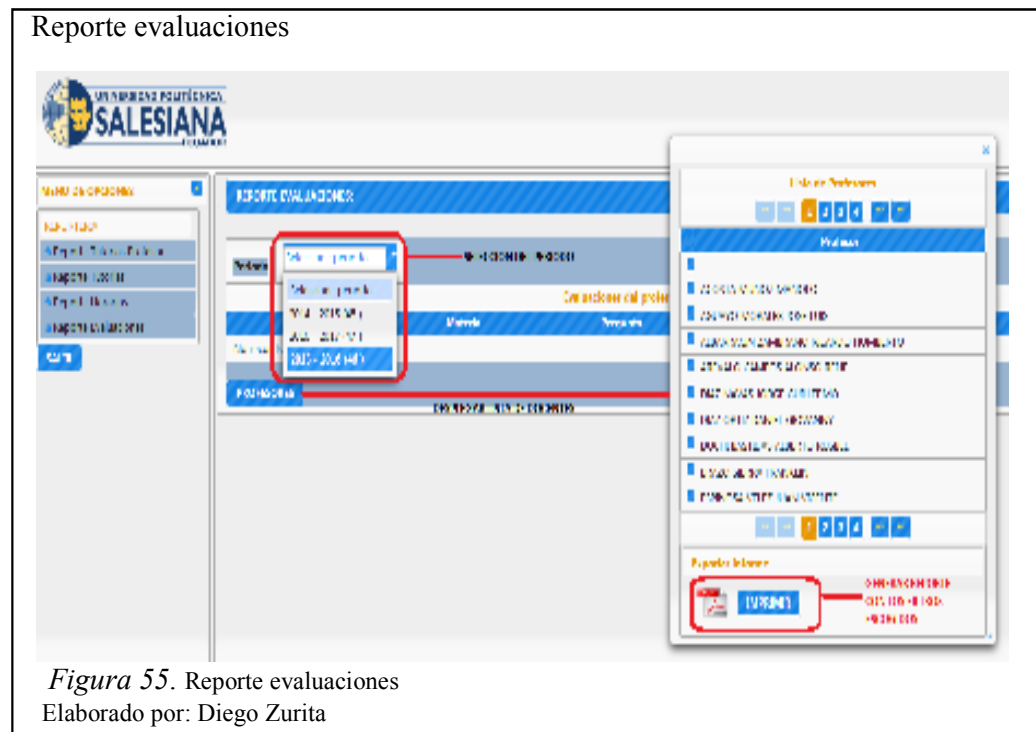
Figura 54. Imagen de reporte de horarios
Elaborado por: Diego Zurita

Reporte Evaluaciones: este reporte mostrará las evaluaciones que ha realizado un docente al impartir una tutoría, y podrá ser filtrado por profesor y período.

Período: selección de los períodos ingresados en el aplicativo sin importar su estado.

Profesores: selección de los profesores que han sido asignados para brindar la tutoría académica.

Botón imprimir: genera el reporte con los filtros seleccionados.



Reporte de gestión

REPORTE DE EVALUACIONES

AREVALO CAMPOS ALONSO RENE

METODOS NUMERICOS

Pregunta	SI	NO
El tutor tiene capacidad para resolver tus	1	0
El tutor tiene	1	0
La cordialidad y capacidad del tutor logra crear un clima de confianza para que	0	1
El tutor te trata con respeto y atención	1	0
Consideras que el programa de tutoría es satisfactorio	1	0

Figura 56. Reporte de calificación tutorías

Elaborado por: Diego Zurita

Reporte Tutorías Profesor: este mostrará todas las tutorías que un docente ha realizado en un período determinado, y podrá ser filtrado por profesor y período.

Período: selección de los períodos ingresados en el aplicativo sin importar su estado.

Profesores: selección de los profesores que han sido asignados para brindar la tutoría académica.

Botón imprimir: genera el reporte con los filtros seleccionados.

Reporte tutorías

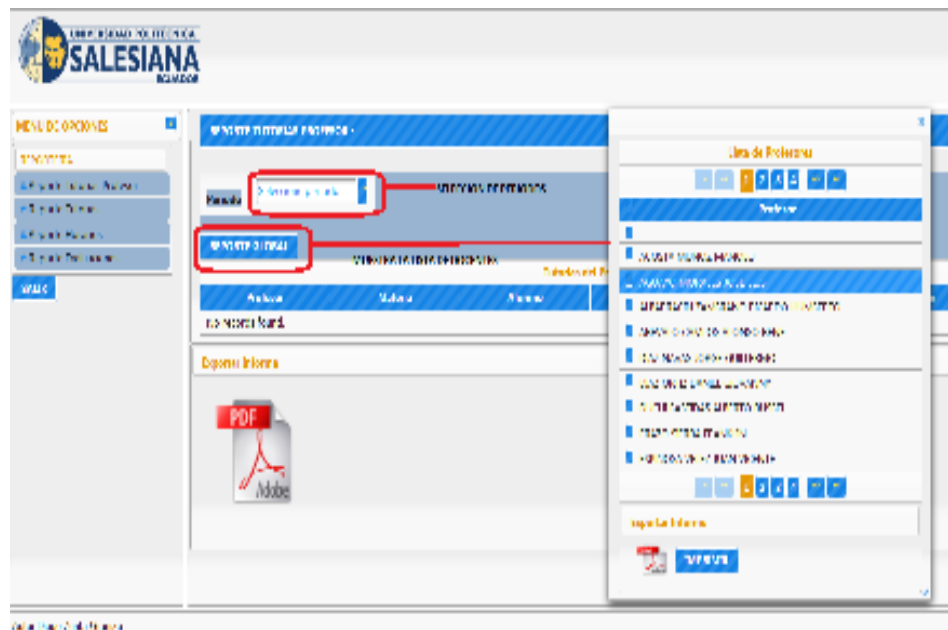


Figura 57. Imagen de la pantalla de filtros del reporte
Elaborado por: Diego Zurita

Reporte de gestión

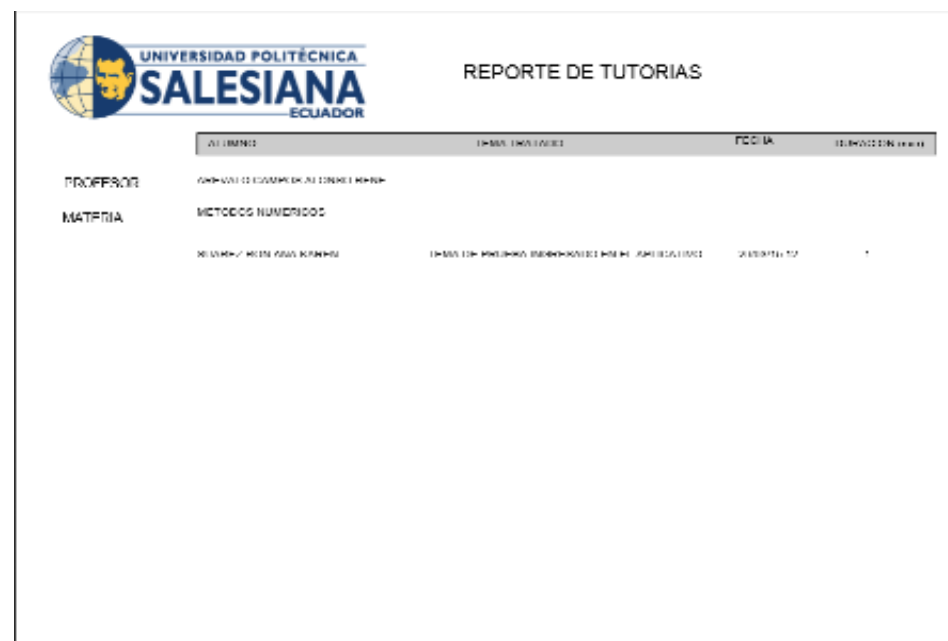


Figura 58. Reporte de tutorías
Elaborado por: Diego Zurita

Reporte Tutorías Alumnos: este mostrará todas las tutorías a las que el estudiante asistió en un período determinado, y podrá ser filtrado por período, materia, grupo y estudiante.

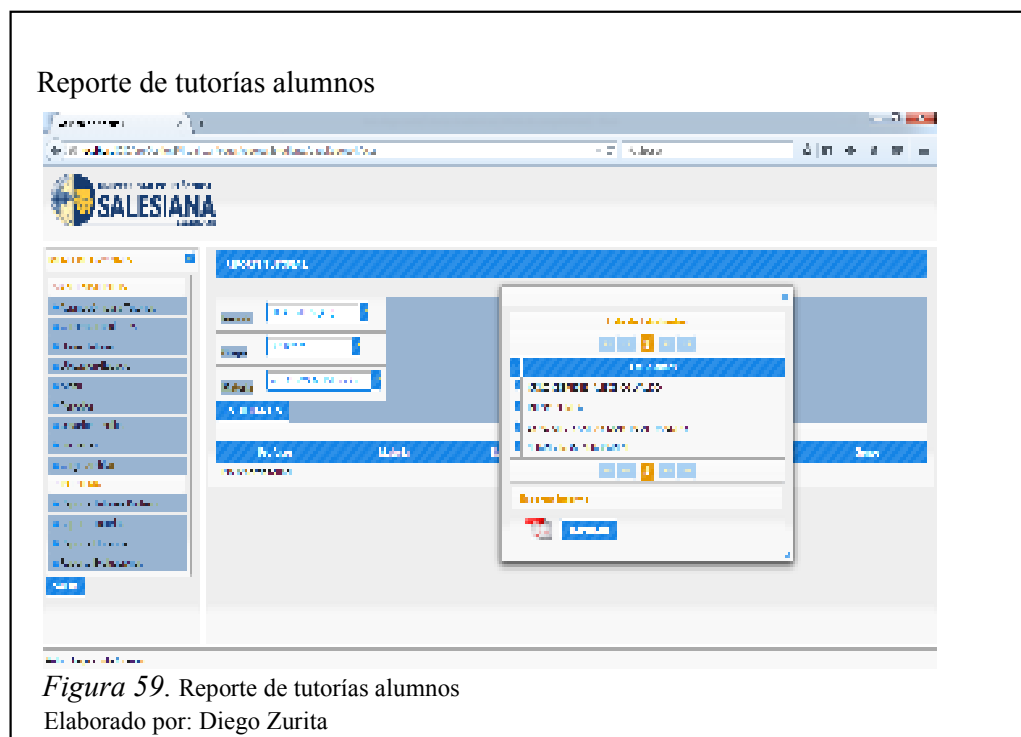
Período: selección de los períodos ingresados en el aplicativo sin importar su estado.

Grupo: selección del grupo al que están ligadas diversas materias.


Materia: selección de las materias ligadas al grupo previamente escogido.

Estudiantes: selección de los profesores que han sido asignados para brindar la tutoría académica.

Botón imprimir: genera el reporte con los filtros seleccionados.



Reporte de gestión



UNIVERSIDAD POLITÉCNICA
SALESIANA
ECUADOR

REPORTE DE TUTORIAS ESTUDIANTE

TEMA	FECHA
------	-------

ESTUDIANTE

SUAREZ ROMANA JORDAN

MATERIA

SEMINARIO DE INVESTIGACION

TEMA DE PRUEBA INGRESADO EN EL APLICATIVO

2015-03-28

Figura 60. Reporte de tutorías estudiantes
Elaborado por: Diego Zurita

3.4 Pruebas

“La prueba es un proceso que se enfoca sobre la lógica interna del software y las funciones externas. Es un proceso de ejecución de un programa con la intención de descubrir un error, no puede asegurar la ausencia de defectos; solo puede demostrar que existen defectos en el software.” (CONSOLACIÓN, 2011)

3.4.1 Pruebas de unidad

“Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados.” (JOSKOWICZ, 2008)

La herramienta utilizada para las pruebas de unidad es HttpUnit, puesto que está orientada al desarrollo web al igual que JUnit es una herramienta de código abierto y la prueba puede correrse en la consola de JUnit.

La prueba realizada a la página inicio.xhtml fue realizada dentro del entorno de netbeans y ejecutada en la consola de JUnit dando como resultado que no posee errores de codificación y el tiempo de carga estimado es de 0,656 segundos.

```

package com.example;

import java.net.URL;

public class STLogin extends BaseClass {
    String url = "http://localhost:8080/jsp/Servlets/Servlets/Login.jsp";

    public void login() {
        try {
            URL url = new URL(url);
            HttpURLConnection connection = new HttpURLConnection();
            HttpURLConnection request = new HttpURLConnection(request.getServerName(), "/");
            System.out.println("request: " + request);
            HttpURLConnection response = connection.getResponse(request);
            System.out.println("response: " + response);
            assertNotNull("url" + url, response);
            System.out.println("url: " + url);

            Map<String, String> map = response.getHeaderFields();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

Figura 61. Prueba de unidad autenticación de usuarios.
Elaborado por: Diego Zurita

La prueba realizada a la página paginatutoria.xhtml de la misma forma fue realizada dentro del entorno de netbeans y ejecutada en la consola de JUnit, dando como resultado que no posee errores de codificación y el tiempo de carga estimada es de 1,251 segundos.

Caso de prueba 2

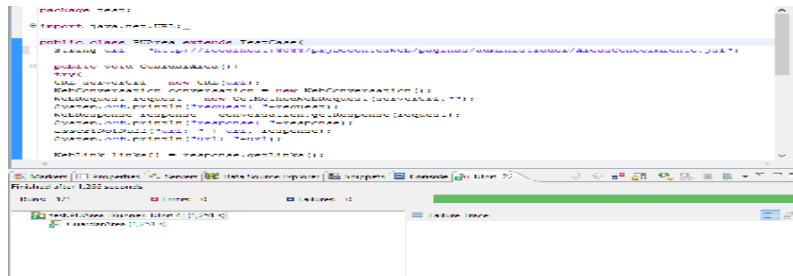


Figura 62. Prueba de unidad ingreso de una tutoría.

Elaborado por: Diego Zurita

Caso de prueba 3: Ingreso de la calificación de la tutoría

La prueba realizada a la página `paginacuestionario.xhtml` de la misma forma fue realizada dentro del entorno de netbeans y ejecutada en la consola de JUnit, dando como resultado que no posee errores de codificación y el tiempo de carga estimada es de 0,703 segundos.

Caso de prueba 3

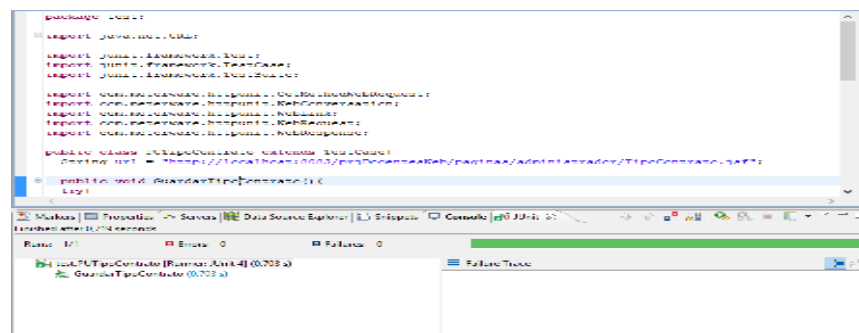


Figura 63. Prueba de unidad calificación de una tutoría.

Elaborado por: Diego Zurita

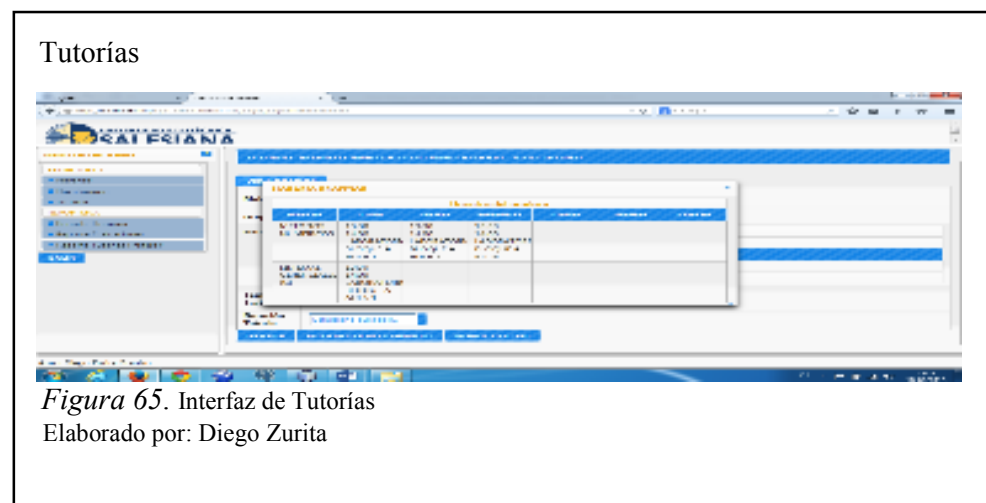
3.4.2 Pruebas de integración

Las pruebas de integración son: “diseñadas para probar la interacción entre los distintos componentes de un sistema”. (UNIVERSIDAD DE SEVILLA, 2013)

Para la realización de la prueba de integración se utilizó el perfil de docente, la primera interfaz seleccionada fue horarios en la cual se ingresó información de un nuevo horario del docente.



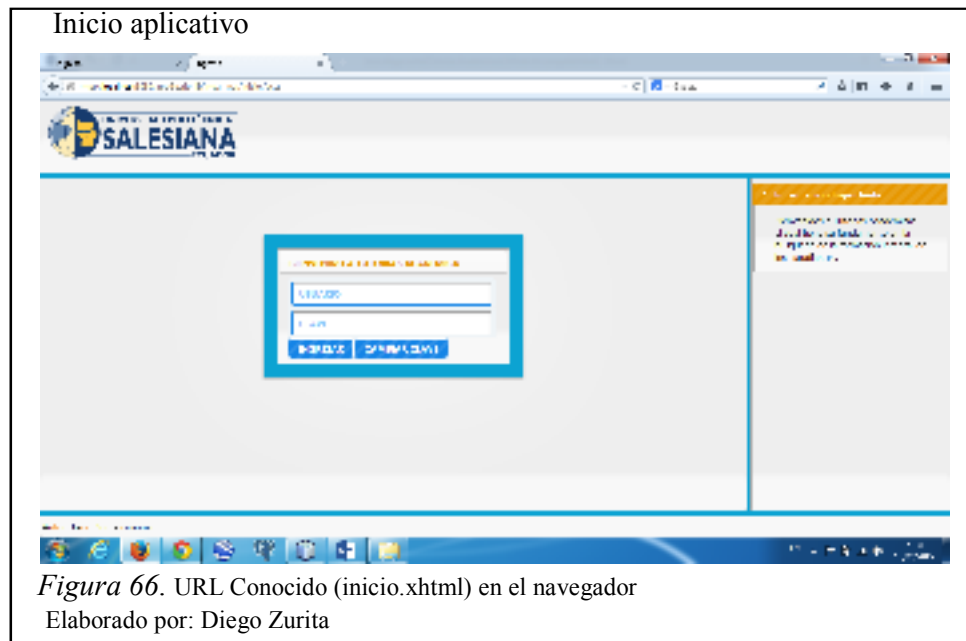
Después del ingreso del horario, el docente puede visualizar esta información cuando está ingresando una tutoría o cuando genera el reporte específico de horarios.



3.4.3 Pruebas de seguridad

Caso de Prueba 1: URL Conocido

Si un usuario conoce un URL y lo coloca en el navegador, directamente el ambiente web direcciona la interfaz del login, puesto que para ingresar, actualizar, eliminar información y ver reportes debe ser un usuario autenticado.



Caso de prueba 2: Caducidad de la sesión

En caso de inactividad del ambiente web por 20 minutos la sesión se caduca y necesariamente debe autenticarse nuevamente para acceder a la información.



Caso de prueba 3: Manejo de perfiles

Si el usuario tiene perfil de docente, estudiante o Director de Carrera puede acceder solo a los módulos que hayan sido asignados a su perfil. El único con acceso a toda la información y reportes es el Administrador.

Manejo de perfiles

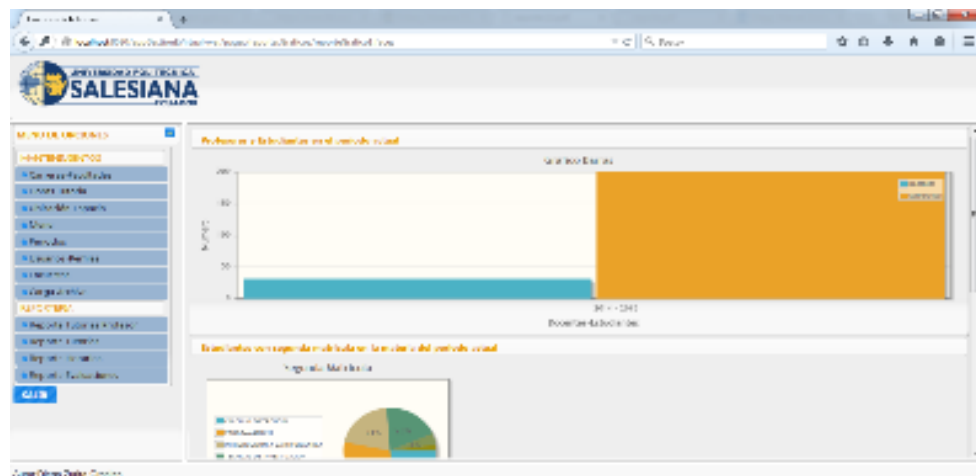


Figura 68. Perfil de administrador
Elaborado por: Diego Zurita

Perfil estudiante

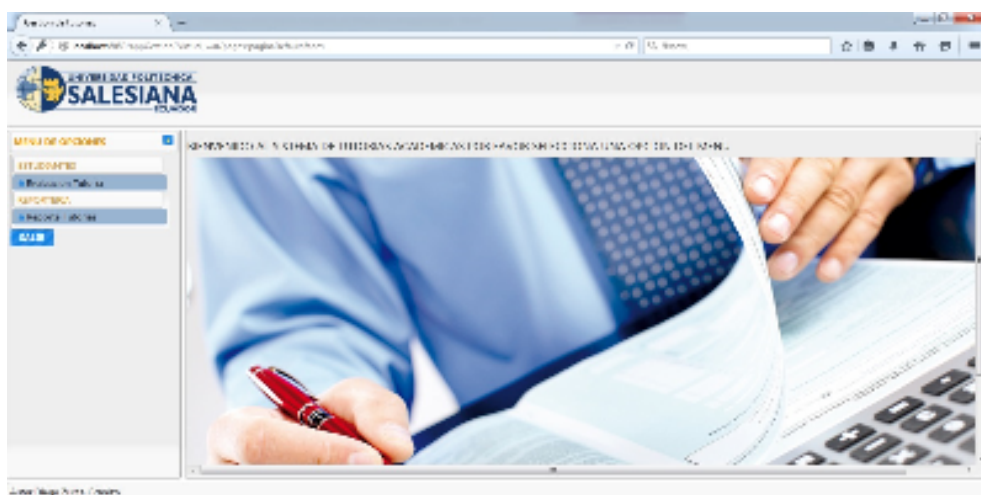


Figura 69. Perfil de estudiante
Elaborado por: Diego Zurita

Perfil docente

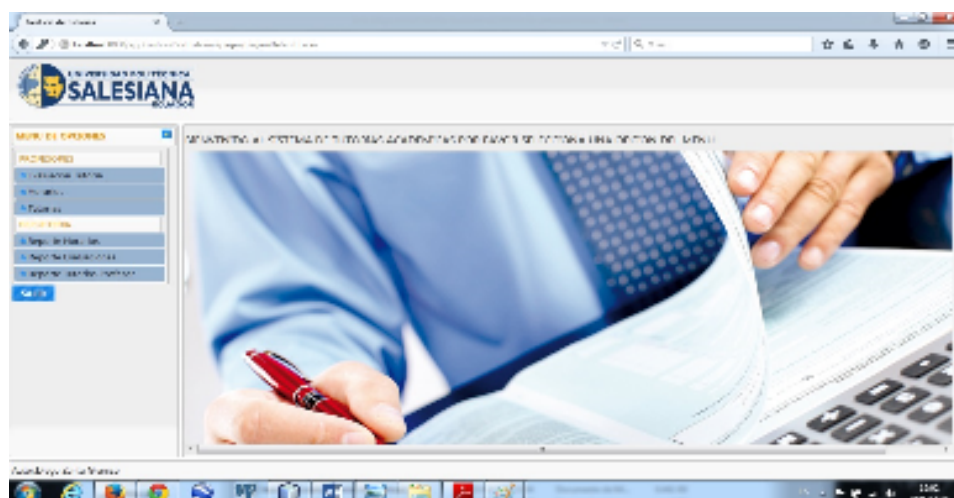


Figura 70. Perfil de docente
Elaborado por: Diego Zurita

Perfil Director de Carrera

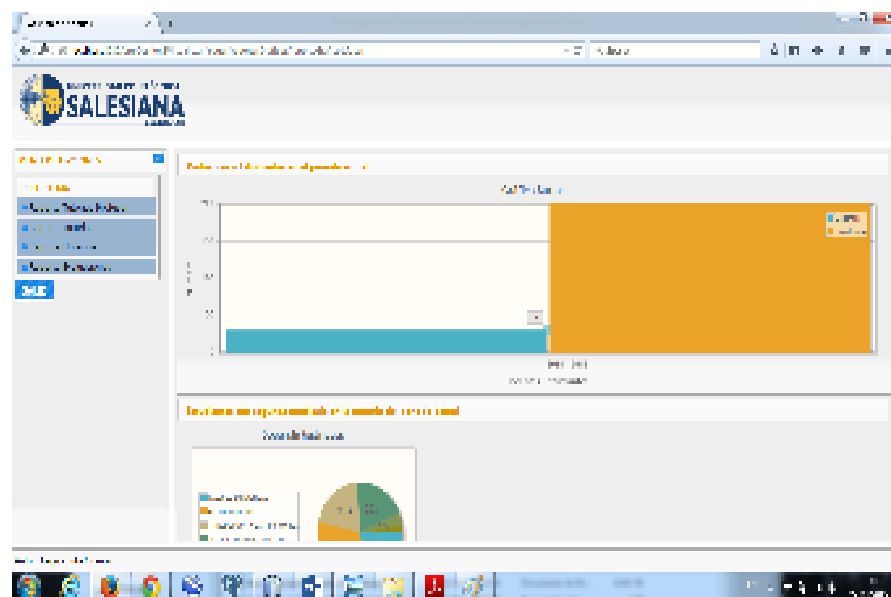


Figura 71. Perfil de Director de Carrera
Elaborado por: Diego Zurita

CONCLUSIONES

- El aplicativo desarrollado permitió el ingreso de información acerca de los estudiantes en riesgo de una manera más ágil y sencilla para iniciar el proceso de la tutoría y brindarles el apoyo académico que necesitan.
- El sistema ha sido creado para que los docentes y estudiantes puedan llevar un registro tutorial inmediato, y que el Director de Carrera pueda consultar los reportes generados de las tutorías impartidas y se pueda tomar las medidas correctivas.
- La metodología XP permitió corregir de forma rápida y eficaz los errores que se presentaron en el desarrollo de esta aplicación y solventar la funcionalidad de los 5 módulos desarrollados.
- La arquitectura de 3 capas permitió tener un mejor desarrollo ordenado y adaptable a cambios en la base de datos o en la implementación de nuevos servicios.
- Esta aplicación permite generar varios reportes, los cuales son de tipo generales que contiene información detallada de varios aspectos de la tutoría, como son: los horarios del docente, las tutorías que se han impartido, la calificación que el estudiante realiza a dicho apoyo académico.
- Los componentes del Primefaces facilitaron el desarrollo de esta aplicación, ya que sus componentes son de fácil manejo para un desarrollador y las interfaces que se pueden desarrollar con esta herramienta son muy amigables con el usuario.

RECOMENDACIONES

- En la siguiente versión de este aplicativo lo recomendable en cuanto al tema de seguridad utilizar la herramienta JAAS con la cual manejaríamos los accesos a los diferentes módulos de esta aplicación y no desde el mantenimiento de menú que se encuentra dentro del aplicativo.
- Es recomendable seguir con el manejo de la metodología XP porque nos facilita todo el desarrollo de la aplicación y nos asegura la completa satisfacción de las necesidades del cliente.
- Es recomendable aumentar las seguridades dentro del aplicativo o dentro del servidor para evitar posibles pérdidas o modificaciones de información con alguna infiltración no deseada.
- El manejo de la herramienta de reportería llamado “iReport” nos facilita la creación y generación de reportes para futuras versiones del aplicativo.
- Es recomendable que para versiones futuras esta aplicación sea configurable en un entorno de dispositivo móvil tomando las debidas seguridades de la utilización de estas librerías.
- Se recomienda el uso de esta aplicación a la Universidad Politécnica Salesiana ya que nos va a permitir tener una visión más clara de cómo se imparten las tutorías a los estudiantes para así poder tomar medias más ágiles y rápidas.
- Se recomienda a los usuarios cotidianos de esta aplicación el manejo responsable de la información.
- Se recomienda que para futuras versiones de esta aplicación se debería obtener la información de los docentes y las materias desde la base de datos de la Universidad Politécnica Salesiana, ya que la misma contiene la información actualizada.

LISTA DE REFERENCIAS

- Almeida, W., & Imbacuán, d. (Mayo de 2009). *Benchmark para el uso de tecnologías relacionadas a servidor de aplicaciones Glassssfish y JBOSS*. Recuperado el Enero de 2014, de <http://bibdigital.epn.edu.ec/bitstream/15000/1513/1/CD-2217.pdf>
- Bahamondes, L. (2 de Noviembre de 2011). Recuperado el 09 de Noviembre de 2013, de www.inf.utfsm.cl/~visconti/xp/Tarjetas_CRC_2.doc
- Calderón, A. (2007). *Metodogías Ágiles*. Recuperado el 02 de Marzo de 2013, de <http://www.seccperu.org/files/Metodologias%20Agiles.pdf>
- Calderón, A. (2007). *Metodologías Ágiles*. Recuperado el 20 de Junio de 2013
- CES. (2012). Reglamento Escalafòn Docente. *RPC-SO-037-NO.265-2012*. Ecuador.
- CONSOLACIÓN, Y. (19 de Marzo de 2011). *Ecuared*. Recuperado el 14 de Enero de 2014, de http://www.ecured.cu/index.php?title=Pruebas_de_software&action=history
- DAO, J. (2014). *Code Futures*. Recuperado el 22 de Abril de 2014, de <http://www.codefutures.com/java-dao/>
- Easten software systems pvt. ltd. (08 de Septiembre de 2006). *Arquitectura de tres capas*. Recuperado el 20 de Abril de 2014, de <http://www.managinf.com/arquitectura.pdf>
- Ferré, x., & Sánchez, m. (2011). *Desarrollo Orientado a Objetos con UML*. México: Grau.
- Freire, J. (2002). *Sistemas Distribuidos Multicapa con Java sobre Linux*. Recuperado el 20 de Abril de 2014, de <http://repositorio.pucesa.edu.ec/jspui/bitstream/123456789/96/1/75207.pdf>
- Gómez, J. (29 de Marzo de 2014). *Juvadi*. Recuperado el 20 de Abril de 2014, de <http://www.judavi.com/n-tier-o-n-layer-acaso-son-la-misma-arquitectura/>
- Guerrero, J. (28 de Junio de 2008). *Metodologías Ágiles de desarrollo de software (XP) Fases*. Recuperado el 03 de Marzo de 2013, de http://boards5.melodysoft.com/UBV_INGS/metodologias-agiles-de-desarrollo-43.html

Instituto Tecnológico Veracruz. (11 de Enero de 2014). *Arquitectura de las aplicaciones Web*. Recuperado el 21 de Abril de 2014, de <http://www.prograweb.com.mx/pweb/0201arquiAplicaweb.html>

Joskowicz, J. (10 de Febrero de 2008). *Reglas y Prácticas en eXtreme Programming*. Recuperado el 22 de Diciembre de 2013, de www.uls.edu.sv/index.php?option=com_phocadownload JPA. (2009). *Ejemplos TIW*. Recuperado el 21 de Abril de 2014, de <http://www.lab.inf.uc3m.es/~a0080802/RAI/jpa.html>

Letelier, p., & Penadés, M. (Junio de 2006). *Cyta*. Recuperado el 24 de Abril de 2013, de http://www.cyta.com.ar/ta0502/b_v5n2a1.htm

Letelier, p., & Sánchez, P. (12 de Noviembre de 2003). *Metodologías ágiles en el desarrollo de software*. Recuperado el 10 de Mayo de 2013, de <http://issi.dsic.upv.es/archives/f1069167248521/actas.pdf>

Oness. (28 de Agosto de 2004). *Ciclo de vida de un proyecto XP*. Recuperado el 20 de Junio de 2013, de Capítulo 5. Metodología: <http://oness.sourceforge.net/proyecto/html/ch05s02.html>

Prieto, p. (10 de Enero de 2013). Características de la aplicación. Quito, Pichincha, Ecuador.

Programación Extrema. (1 de Septiembre de 2004). Recuperado el 15 de Junio de 2013, de <http://programacionextrema.tripod.com/fases.htm>

Schmuller, J. (2001). *Aprendiendo UML en 24 horas*. Estados Unidos, Schumuller Joseph: Pearson Education.

Schmuller, J. (2001). *Aprendiendo UML en 24 horas*. Estados Unidos.

SNA, S. N. (14 de Enero de 2013). Datos del número de alumnos en la carrera de Sistemas. Quito, Pichincha, Ecuador.

Toala, G. (20 de febrero de 2012). Resumen de la Historia de la Carrera. Quito, Pichincha, Ecuador.

Universidad de Guayaquil. (13 de Agosto de 2008). *Programación por capas*. Recuperado el 21 de Abril de 2014, de oasis.ciscug.org/letzhune/cisc/.../Programacion%20por%20capas.doc

Universidad de Sevilla. (19 de Mayo de 2013). *Integración de Sistemas Software*. Recuperado el 15 de Enero de 2014, de Pruebas de Integración: <http://www.lsi.us.es/docencia/get.php?id=6731>

Universidad de Talca. (21 de Diciembre de 2013). *API Java de persistencia*. Recuperado el 22 de Abril de 2014, de http://convenios.otalca.cl/actas/descargar_acta.php?token=&mod=1&nombre=Persistencia.pdf.

Universidad Politécnica Salesiana. (s.f.). *Universidad Politécnica Salesiana* . Recuperado el 15 de Diciembre de 2013, de <http://www.ups.edu.ec/> Zuluaga, C. (Diciembre de 2008). *Enterprise Architect y UML Básico*. Recuperado el 16 de Enero de 2014, de <http://carloszuluaga.wdfiles.com/local--files/cursos-talleres:enterprisearchitect/CursoBasicoEA-Sesion07.pdf>

ANEXOS

El formato con el cual el docente debe publicar los horarios en los cuales impartirá las tutorías es el siguiente

Anexo 1. Formato de la publicación horarios tutoría

CARRERA DE INGENIERÍA DE SISTEMAS
HORARIO DE TUTORIAS

DOCENTE:

OFICINA:

LUGAR DE LAS TUTORIAS:

SELECCIONAR EL NOMBRE DEL DOCENTE

MATERIA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO
ELEGIR LA MATERIA O DIRECCIÓN DE TESIS						
ELEGIR LA MATERIA O DIRECCIÓN DE TESIS						
ELEGIR LA MATERIA O DIRECCIÓN DE TESIS						
ELEGIR LA MATERIA O DIRECCIÓN DE TESIS						
ELEGIR LA MATERIA O DIRECCIÓN DE TESIS						
ELEGIR LA MATERIA O DIRECCIÓN DE TESIS						
ELEGIR LA MATERIA O DIRECCIÓN DE TESIS						
ELEGIR LA MATERIA O DIRECCIÓN DE TESIS						
ELEGIR LA MATERIA O DIRECCIÓN DE TESIS						

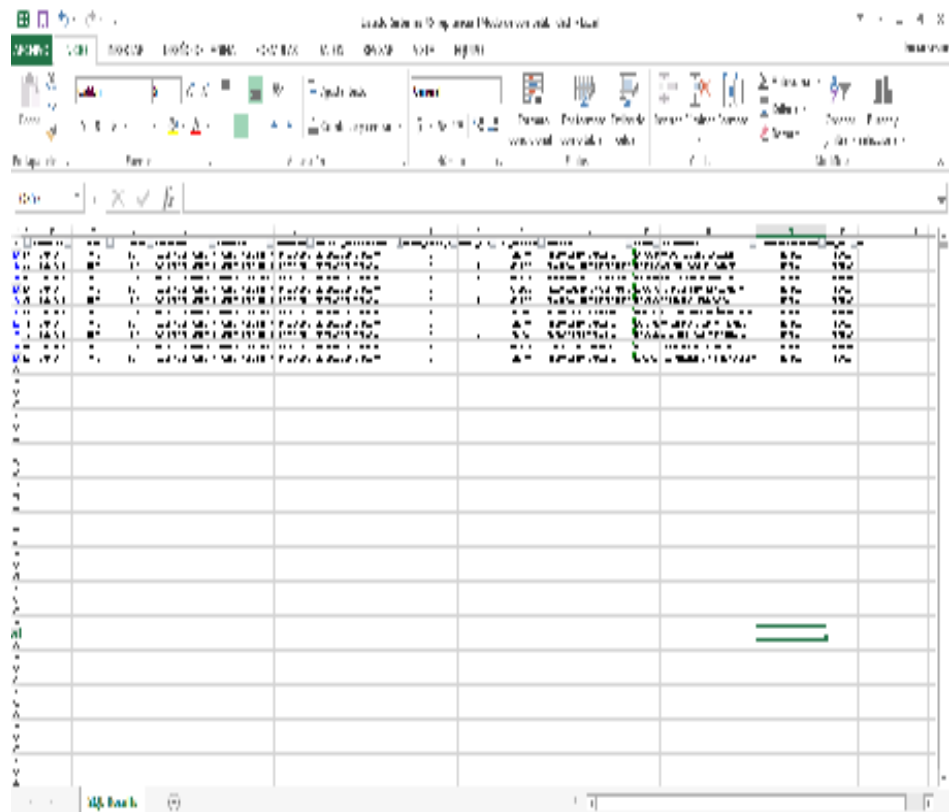
Anexo 2. Registro del tutor de asistencia a tutorías

NOMBRE DEL TUTOR								
ASIGNATURA								
PERÍODO (semestre)								
HORARIO DE LAS TUTORÍAS								
No	Nombre Estudiante (Tutoriado)	Nivel	Paralel	No Mat rícula	Tema Tratado	Fecha De La Tutoría DD/MM/AA	Firma Estudiante	

Anexo 3. Registro del tutorado

NOMBRE DEL ESTUDIANTE-TUTORIADO			
ASIGNATURA			
PERÍODO (SEMESTRE)			
NO DE MATRÍCULA			
No	Fecha de la tutoría DD/MM/AA	Tema tratado	Firma tutor
1			
2			
3			
4			
5			
6			
7			
8			
9			

Anexo 4. Formato del archivo a subir al sistema



The screenshot shows a Microsoft Excel spreadsheet with the following structure:

- Columns:** The spreadsheet has 10 columns. The first column contains row numbers (1 to 10). The second column contains a header 'ID'. The third column contains a header 'Nombre'. The fourth column contains a header 'Apellido'. The fifth column contains a header 'Fecha de nacimiento'. The sixth column contains a header 'Sexo'. The seventh column contains a header 'E-mail'. The eighth column contains a header 'Telefono'. The ninth column contains a header 'Direccion'. The tenth column contains a header 'Ciudad'.
- Rows:** The spreadsheet has 10 rows. The first row is the header row. The second row contains the following data: '1', 'ID', 'Nombre', 'Apellido', 'Fecha de nacimiento', 'Sexo', 'E-mail', 'Telefono', 'Direccion', 'Ciudad'. The third row contains the following data: '2', '1', 'Juan', 'Perez', '1990-01-01', 'M', 'juan.perez@ejemplo.com', '987654321', 'Calle 123', 'Bogota'.